

KENDRIYA VIDYALAYA NO.1

GANDHI NAGAR, JAMMU



तत् त्वं पूषन् अपावृणु
केन्द्रीय विद्यालय संगठन

A PROJECT REPORT

ON

“CONTACT MANAGEMENT SYSTEM”

FOR

AISSCE 2016 EXAMINATION

As a part of the Informatics Practices Course (065)

SUBMITTED BY:

SABREESH BHATT

ROLL NO- 8

BOARD ROLL NO –

Under the Guidance of:

MR. SURJEET SINGH

PGT(CS)

CERTIFICATE

This is to certify that the project/dissertation entitled, CONTACT MANAGEMENT SYSTEM is a bonafide work done by Mr. Sabreesh Bhatt of class 12th A Session 2015-16 in partial fulfilment of CBSE's AISSCE Examination 2016 and has been carried out under my direct supervision and guidance. This report or a similar report on the topic has not been submitted for any other examination and does not form part of any other course undergone by the candidate.

.....

Signature of Student

Name:

Roll no:

.....

Signature of Teacher/Guide

Name:

Designation:

.....

Signature of Principal

Name:

Place:

Date:

ACKNOWLEDGEMENT

I sincerely thank our IP Teacher Mr. Surjeet Singh for his guidance, encouragement and support throughout the duration of the project. Without his motivation and help the successful completion of this project would not have been possible. I would also like to extend my gratitude to the entire IP department of Kendriya Vidyalaya No. 1, Jammu for their constant support.

Sabreesh Bhatt

12th A

(Name and signature of the student)

INDEX

1. Certificate
2. Acknowledgement
3. Objective and scope of the project
4. Theoretical Background
5. Definition of Problem
6. System Planning
7. Methodology Adopted, Details of Hardware And Software Used
8. Interface Design and Coding of Methods
9. Structure of Tables
10. Test Report
11. User/Operational Method
12. Brief background of the organization
13. Reference

OBJECTIVE AND SCOPE OF THE PROJECT

The goal of this project is to create a contact management system or contact manager which will store all the information of user and enable the user to access that information from it whenever required.

This project has a very wide scope and it can provide the following advantages:

- . Centralized repository of contact information
- . Ready to use database with searching
- . Sales tracking
- . Email integration
- . Scheduling of appointments and meetings
- . Document management
- . Notes and conversation management
- . Customizable fields
- . Drip marketing
- . Import/export utility

THEORETICAL **BACKGROUND**

Contact Manager is an ideal solution for small teams and business professionals who need a simple solution to manage their contacts and help them with day-to-day business activities. An insurance agent or a music composer, a start-up with a team of 5 or a fashion boutique for instance, Contact Manager would be a perfect fit. They might try to use CRM or an address book. But then

CRM is overkill

Software is one area, where bigger is not necessarily better. If you find traditional CRM systems to be overwhelming for your needs, then Contact Manager might be a better fit for you.

Address Book is too primitive

You can use an address book mostly to store contacts. Expecting more out of it wouldn't be fair. But Contact Manager can help you with other stuff like tasks, deals, email and team collaboration apart from managing your business contacts.

DEFINITION OF **PROBLEM**

Contact Management System is a must needed software and it was chose to overcome the problems related to storage, access, modification and deletion of contacts smoothly.

Earlier it was very difficult for public to do all these tasks smoothly and comfortably and that is when the concept of the need of CMS arise. Due to the difficulties faced, it was very important to solve all these problems, so that we can save time and do all these tasks without errors.

SYSTEM PLANNING

The planning of this project had proceeded in the following sequence:

- Basic Idea about Contact Management System
- Outline Imagination for all the frames
- Collection of Data required for Project
- Searching for Required help on the internet and from the subject teacher
- Elements required in each frame
- Tables Required with their attributes
- Coding on all the buttons

METHODOLOGY **ADOPTED, DETAILS** **OF HARDWARE AND** **SOFTWARE USED**

Methodology Adopted in the making of this project was as follows:

- Planning of the Structure of the project
- Formation of the frames
- Formation of tables
- Coding in Each Frame and connectivity with database
- Collaboration of all the frames into one major project
- Adding final touch to all the frames

Hardware Required for the successful run of the project:

- 1 GHZ Processor
- 1 GB RAM or Above
- Intel Pentium dual core processor or above

Software Required for the successful run of the project:

- NetBeans 7.0 or Above
- MySQL 5.1 or Above
- Java Development Kit(jdk)
- Windows Vista or Above

INTERFACE DESIGN **AND CODING OF** **METHODS**

1. Contact Management System ***Screen***

Welcome To Contact Management System...

Enter To The DataBase

Cancel

CODING:

BUTTON--ENTER TO DATABASE

```
MainEntryToThePregramme loginform=new MainEntryToThePregramme();  
    loginform.setVisible(false);  
        setVisible(false);
```

BUTTON--CANCEL

```
JOptionPane.showMessageDialog(null,"Thank you for visiting our site!! and  
    to exit please click on the X mark on the upper right corner..");
```

2. Main Entry To The Form



CODING:

BUTTON--LOGIN

```
String pwd=new String (pwdFld.getPassword());  
    if (pwd.equals("pranav"))  
    {  
        Msg.setText("Password Valid!");  
        MainForm loginform=new MainForm();  
  
        loginform.setVisible(true);  
        setVisible(false);  
    }
```

```
else  
{  
JOptionPane.showMessageDialog(null,"Invalid Password..?");  
}
```

BUTTON--EXIT

```
JOptionPane.showMessageDialog(null," Thanks For Using Database  
System");  
System.exit(0);
```

3. Main Form

The image shows a graphical user interface for a 'Main Form'. It has a light gray background with a thin blue border. At the top, the text 'Events Included!' is centered. Below this, there are five rectangular buttons stacked vertically, each with a thin gray border. The buttons are labeled 'Add Contacts', 'Update/Delete Contact', 'Search Contact', 'Back', and 'Exit' from top to bottom. The 'Exit' button is smaller and positioned further down than the others.

CODING:

BUTTON—UPDATE/DELETE CONTACT

```
UpdateDeleteForm loginform=new UpdateDeleteForm();  
loginform.setVisible(true);  
setVisible(false);
```

BUTTON—ADD CONTACTS

```
AddContactForm loginform=new AddContactForm();
```

```
loginform.setVisible(true);  
setVisible(false);
```

BUTTON-SEARCH CONTACTS

```
SearchContactForm loginform=new SearchContactForm();  
loginform.setVisible(true);  
setVisible(false);
```

BUTTON--BACK

```
MainEntryToTheProgramme loginform=new MainEntryToTheProgramme();  
loginform.setVisible(true);  
setVisible(false);
```

BUTTON--EXIT

```
System.exit(0);
```


4. Inner Structure

A. IN “ADD CONTACTS”

Events Inc.

Menu For Adding Corresponding Details

<i>Add Address</i>	<i>Add Phones</i>
<i>Add E-Mail Address</i>	<i>Add BirthDay</i>
<i>Add Anniversary</i>	<i>Add Remainder</i>

Back

• ***ADD ADDRESS***

Events Inc.
Address Entry Form

Enter Id:

Name:

Address:

City:

State:

Country

PinCode:

New Record

Add Record

Back

CODING:

BUTTON—NEW RECORD

```
Entry_id.setText("");
Person_Name.setText("");
address.setText("");
city.setText("");
state.setText("");
country.setText("");
zipcode.setText("");
```

BUTTON—ADD RECORD

```
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://
localhost:3306/cms","root","");
Statement stmt=con.createStatement();
String id =Entry_id.getText();
String name= Person_Name.getText();
String addr =address.getText();
String cty =city.getText();
String st =state.getText();
String cntry =country.getText();
String zp =zipcode.getText();
```

```
String str="insert into address(id,name,address,city,state,country,pincode)
        values('"+(id)+"','"+(name
)+"', '"+(addr)+"', '"+(cty)+"', '"+(st)+"', '"+(cntry)+"', '"+(zp)+"')";
        int r= stmt.executeUpdate(str);
        JOptionPane.showMessageDialog(this,"Record inserted"+r);
```

```
        }
        catch(Exception e)
        {
        JOptionPane.showMessageDialog(this,e.getMessage());
        e.printStackTrace();
        }
```

BUTTON--BACK

```
AddContactForm loginform=new AddContactForm();
        loginform.setVisible(true);
        setVisible(false);
```

• ***ADD EMAIL ADDRESS***

Events Included!

E-MAIL ENTRY ENTRY FORM

E-Mail ID:

Name:

Alternate Email_ID

CODING:

BUTTON—ADD RECORD

```
try
{
Class.forName("com.mysql.jdbc.Driver");

Connection con= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

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

```
String emid =jTextField2.getText();
```

```
String n= jTextField3.getText();
```

```
String alteremail =jTextField1.getText();
```

```
String str="insert into email(email,name,alteid)  
values('"+(emid)+"','"+(n)+"','"+(alteremail)+"");
```

```
int r= stmt.executeUpdate(str);
```

```
JOptionPane.showMessageDialog(this,"Record inserted"+r);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

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

```
}
```

BUTTON--BACK

```
AddContactForm loginform=new AddContactForm();
```

```
loginform.setVisible(true);
```

```
setVisible(false);
```

• ***ADD ANNIVERSARY***

Events Included!

ANNIVERSARY ENTRY ENTRY FORM

Enter ID:

Name:

Enter Date Of Anniversary: Day: ▾ Month: ▾ Year:

CODING:

BUTTON-SAVE

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
```

```

        int id =Integer.parseInt(jTextField1.getText());
        String n= jTextField2.getText();

        int d =Integer.parseInt(jComboBox1.getSelectedItem().toString());
        int m =Integer.parseInt(jComboBox2.getSelectedItem().toString());
        int yr=Integer.parseInt( jTextField3.getText());

        String str="insert into aniversry(id,name,day,month,year)
        values('"+(id)+"','"+(n)+"','"+(d)+"','"+(m)+"','"+(yr)+"')";

        int r= stmt.executeUpdate(str);

        JOptionPane.showMessageDialog(this,"Record inserted"+r);
    }

    catch(Exception e)
    {

        JOptionPane.showMessageDialog(this,e.getMessage());
        e.printStackTrace();
    }

```

BUTTON--BACK

```

AddContactForm loginform=new AddContactForm();
loginform.setVisible(true);
setVisible(false);

```


• ***ADD PHONES***

Events Included!
PHONE NO. ENTRY FORM

E-Mail ID:

Name:

Phone Number:

CODING:

BUTTON—ADD RECORD

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

```
Statement stmt=con.createStatement();
String em =jTextField1.getText();
String name= jTextField2.getText();
int ph =Integer.parseInt(jTextField3.getText());

String str="insert into phone(email,name,phno)
values('"+(em)+"','"+(name )+"','"+(ph)+"')";
int r= stmt.executeUpdate(str);
JOptionPane.showMessageDialog(this,"Record inserted"+r);
}
```

```
catch(Exception e)
{
JOptionPane.showMessageDialog(this,e.getMessage());
e.printStackTrace();
}
```

BUTTON--BACK

```
AddContactForm loginform=new AddContactForm();
loginform.setVisible(true);
setVisible(false);
```

• ***ADD BIRTHDAY***

Events Included!

BIRTHDAY ENTRY ENTRY FORM

Enter ID:

Name:

Enter Date Of Birth: Day: Month: Year:

CODING:

BUTTON-SAVE

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    int id =Integer.parseInt(jTextField1.getText());
```

```

        String n= jTextField2.getText();

        int d =Integer.parseInt(jComboBox1.getSelectedItem().toString());
        int m =Integer.parseInt(jComboBox2.getSelectedItem().toString());

        int yr=Integer.parseInt( jTextField3.getText());

        String str="insert into birthday(id,name,day,month,year)
        values('"+(id)+"','"+(n)+"','"+(d)+"','"+(m)+"','"+(yr)+"')";

        int r= stmt.executeUpdate(str);

        JOptionPane.showMessageDialog(this,"Record inserted"+r);

    }

    catch(Exception e)

    {

        JOptionPane.showMessageDialog(this,e.getMessage());

        e.printStackTrace();

    }

```

BUTTON--BACK

```

AddContactForm loginform=new AddContactForm();

        loginform.setVisible(true);

        setVisible(false);

```

• ***ADD REMAINDER***

Events Included!

REMAINDER ENTRY ENTRY FORM

Enter ID:

Name:

Enter Remainder Date:

Enter Date Of Birth: Day: Month: Year:

CODING:

BUTTON-SAVE

```
try
```

```
{
```

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

```
Connection con= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```

        Statement stmt=con.createStatement();
        int id =Integer.parseInt(jTextField1.getText());
        String n= jTextField2.getText();
        String rd=jTextField4.getText();
        int d =Integer.parseInt(jComboBox3.getSelectedItem().toString());
        int m =Integer.parseInt(jComboBox4.getSelectedItem().toString());
        int yr=Integer.parseInt( jTextField3.getText());
        String str="insert into remainder(id,name,rem_date,day,month,year)
        values('"+(id)+"','"+(n)+"','"+(rd)+"','"+(d)+"','"+(m)+"','"+(yr)+"')";
        int r= stmt.executeUpdate(str);
        JOptionPane.showMessageDialog(this,"Record inserted"+r);
    }
    catch(Exception e)
    {
        JOptionPane.showMessageDialog(this,e.getMessage());
        e.printStackTrace();
    }

```

BUTTON--BACK

```

AddContactForm loginform=new AddContactForm();
loginform.setVisible(true);
setVisible(false);

```

***B. IN “UPDATE/DELETE
CONTACTS”***

Events Included!

Menu For Updating/Delete Corresponding Details

Update/Delete Address

Update/Delete Phone No

Update/Delete E-Mail

Update/Delete Birthday

Update/Delete Anniversary

Update/Delete Remainder

Back

• **UPDATE/DELETE PHONE NO.**

Events Included!

Phone Updation/Deletetion Form

Select Name: ▼

Email ID:

Name:

phone:

CODING:

BUTTON-DELETE

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con1= (Connection)
```



```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();
```

```
String n=(String)jComboBox1.getSelectedItem();
```

```
String str1="delete from phone where name='"+(n)+"'";
```

```
ResultSet rs1;
```

```
int r = stmt1.executeUpdate(str1);
```

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

```
jComboBox1.removeItem(n);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

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

```
}
```

BUTTON—UPDATE

```
try
```

```
{
```

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

```
Connection con1= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();
```

```

String n=(String)jComboBox1.getSelectedItem();
String em=jTextField1.getText();
String name=jTextField2.getText();

String ph=jTextField3.getText();
String str2="update phone set name='"+(name)+"',
email='"+(em)+"',phno='"+(ph)+"' where name='"+(n)+"'";

ResultSet rs2;
int r = stmt1.executeUpdate(str2);

JOptionPane.showMessageDialog(this,"Record Updated");
    }
    catch(Exception e)
    {
JOptionPane.showMessageDialog(this,e.getMes sage());
e.printStackTrace();
    }

```

BUTTON--BACK

```

UpdateDeleteForm loginform=new UpdateDeleteForm();
loginform.setVisible(true);
setVisible(false);

```

• **UPDATE/DELETE EMAIL**

Events Included!

E-Mail Updation/Deletetion Form

Select Entry: Item 1 ▾

Delete Update

Enter ID:

Name:

E-Mail ID:

Back

CODING:

BUTTON-DELETE

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con1= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();
```

```
String n=(String)jComboBox1.getSelectedItem();
```

```
String str1="delete from email where name='"+n+"'";
```

```
ResultSet rs1;
```

```
int r = stmt1.executeUpdate(str1);
```

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

```
jComboBox1.removeItem(n);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

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

```
}
```

BUTTON—UPDATE

```
try
```

```
{
```

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

```
Connection con1= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();  
String n=(String)jComboBox1.getSelectedItem();  
String em=jTextField1.getText();
```

```
String name=jTextField2.getText();  
String str2="update email set name='"+(name)+"', email='"+(em)+"' where  
name='"+(n)+"'";
```

```
ResultSet rs2;  
int r = stmt1.executeUpdate(str2);
```

```
JOptionPane.showMessageDialog(this,"Record Updated");  
}
```

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

```
JOptionPane.showMessageDialog(this,e.getMes sage());
```

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

BUTTON--BACK

```
UpdateDeleteForm loginform=new UpdateDeleteForm();  
loginform.setVisible(true);  
setVisible(false);
```

• *UPDATE/DELETE BIRTHDAY*

Events Included!

UPDATE/DELETE BIRTHDAY FORM

Select Contact:

Enter ID:

Name:

Enter Date Of Anniversary: Day: Month: Year:

CODING:

BUTTON-DELETE

```
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con1= (Connection)
DriverManager.getConnection("jdbc:mysql://
localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();  
String n=(String)jComboBox1.getSelectedItem();  
String str1="delete from birthday where name='"+(n)+"'";
```

```
        ResultSet rs1;  
        int r = stmt1.executeUpdate(str1);  
        System.out.println(r);  
        jComboBox1.removeItem(n);  
    }  
    catch(Exception e)  
    {  
JOptionPane.showMessageDialog(this,e.getMessage());  
        e.printStackTrace();  
    }
```

BUTTON—UPDATE

```
        try  
        {  
            Class.forName("com.mysql.jdbc.Driver");  
            Connection con1= (Connection)  
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");  
            Statement stmt1=con1.createStatement();  
            String n=(String)jComboBox1.getSelectedItem();
```

```
String em=jTextField1.getText();
```

```
String name=jTextField2.getText();
```

```
int d =Integer.parseInt(jComboBox1.getSelectedItem().toString());
```

```
int m =Integer.parseInt(jComboBox2.getSelectedItem().toString());
```

```
int y=Integer.parseInt( jTextField3.getText());
```

```
String str2="update birthday set name='"+(name)+"',d='"+(d)+"'
,m='"+(m)+"',y='"+(y)+"'where name='"+(n)+"'";
```

```
ResultSet rs2;
```

```
int r = stmt1.executeUpdate(str2);
```

```
JOptionPane.showMessageDialog(this,"Record Updated");
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMes sage());
```

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

BUTTON—BACK

```
UpdateDeleteForm loginform=new UpdateDeleteForm();
```

```
loginform.setVisible(true);
```

```
setVisible(false);
```


• **UPDATE/DELETE ANNIVERSARY**

Events Included!

UPDATE/DELETE ANNIVERSARY FORM

Select Contact:

Enter ID:

Name:

Enter Date Of Anniversary: Day: Month: Year:

CODING:

BUTTON-DELETE

```
try
{
Class.forName("com.mysql.jdbc.Driver");
Connection con1= (Connection)
DriverManager.getConnection("jdbc:mysql://
localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();  
String n=(String)jComboBox1.getSelectedItem();  
String str1="delete from aniversry where name='"+(n)+"'";
```

```
ResultSet rs1;  
int r = stmt1.executeUpdate(str1);  
System.out.println(r);  
jComboBox1.removeItem(n);  
}  
catch(Exception e)  
{  
JOptionPane.showMessageDialog(this,e.getMessage());  
e.printStackTrace();  
}
```

BUTTON—UPDATE

```
try  
{  
Class.forName("com.mysql.jdbc.Driver");  
Connection con1= (Connection)  
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");  
Statement stmt1=con1.createStatement();  
String n=(String)jComboBox1.getSelectedItem();  
String em=jTextField1.getText();
```

```
String name=jTextField2.getText();  
int d =Integer.parseInt(jComboBox1.getSelectedItem().toString());  
int m =Integer.parseInt(jComboBox2.getSelectedItem().toString());  
int y=Integer.parseInt( jTextField3.getText());
```

```
String str2="update aniversary set name='"+(name)+"',d='"+(d)+"'  
,m='"+(m)+"',y='"+(y)+"'where name='"+(n)+"'";
```

```
ResultSet rs2;  
int r = stmt1.executeUpdate(str2);  
  
JOptionPane.showMessageDialog(this,"Record Updated");  
    }  
catch(Exception e)  
{  
JOptionPane.showMessageDialog(this,e.getMessage());  
e.printStackTrace();  }
```

BUTTON—BACK

```
UpdateDeleteForm loginform=new UpdateDeleteForm();  
loginform.setVisible(true);  
setVisible(false);
```

• UPDATE/DELETE REMAINDER

Events Included!
UPDATE/DELETE REMAINDERS FORM

Select Contact:

Enter ID:

Name:

Enter Remainers Text:

Enter Date Of Anniversary: Day: Month: Year:

CODING:

BUTTON—DELETE

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con1= (Connection)
    DriverManager.getConnection("jdbc:mysql://
    localhost:3306/cms","root","");
    Statement stmt1=con1.createStatement();
    String n=(String)jComboBox1.getSelectedItem();
```

```
String str1="delete from aniversry where name='"+(n)+"'";
```

```
ResultSet rs1;
```

```
int r = stmt1.executeUpdate(str1);
```

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

```
jComboBox1.removeItem(n);
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

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

```
}
```

BUTTON—UPDATE

```
try
```

```
{
```

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

```
Connection con1= (Connection)
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
```

```
Statement stmt1=con1.createStatement();
```

```
String n=(String)jComboBox1.getSelectedItem();
```

```
String em=jTextField1.getText();
```

```
String name=jTextField2.getText();
```

```
String rem=jTextField3.getText();  
int d =Integer.parseInt(jComboBox1.getSelectedItem().toString());  
int m =Integer.parseInt(jComboBox2.getSelectedItem().toString());  
int y=Integer.parseInt( jTextField4.getText());
```

```
String str2="update remainder set  
name='"+(name)+"',rem='"+(rem)+"',d='"+(d)+"'  
,m='"+(m)+"',y='"+(y)+"'where name='"+(n)+"'";
```

```
ResultSet rs2;  
int r = stmt1.executeUpdate(str2);  
  
JOptionPane.showMessageDialog(this,"Record Updated");  
    }  
catch(Exception e)  
    {  
JOptionPane.showMessageDialog(this,e.getMes sage());  
  
e.printStackTrace();  }
```

BUTTON—BACK

```
UpdateDeleteForm loginform=new UpdateDeleteForm();  
loginform.setVisible(true);  
setVisible(false);
```

C. IN “SEARCH CONTACTS”

Events Inc.

Menu For Searching Corresponding Details

Search Address

Search E-Mail

Search Phone.No

Search BirthDay

Search Anniversary

Search Remainder

Back

• ***SEARCH ADDRESS***

Events Inc.
Address Details Form

Name

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    String id =jTextField1.getText();
```



```

String str="Select * from address where name='"+id+"'";
ResultSet s=stmt.executeQuery(str);
    while (s.next())
    {String id1=s.getString("id");
        String uname=s.getString("name");
        String uaddress=s.getString("address");
        String ucity=s.getString("city");
        String ustate=s.getString("state");
        String ucountry=s.getString("country");
        String upincode=s.getString("pincode");

        jLabel4.setText("ID : "+id1+" Name is : "+uname+" City is : 
"+ucity+" State is : "+ustate+" Country is : "+ucountry+" Pincode is : 
"+upincode+"\n"); }

    }

    catch(Exception e)

    {JOptionPane.showMessageDialog(this,e.getMessage());
        e.printStackTrace();}

```

BUTTON—BACK

```

SearchContactForm loginform=new SearchContactForm();

loginform.setVisible(true);

setVisible(false);

```

• ***SEARCH E-MAIL***

Events Inc.
E-Mail Details Form

Name

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
```

```

        String id = jTextField1.getText();
        String str="Select * from email where name='"+id+"'";
        ResultSet s=stmt.executeQuery(str);
        while (s.next())
        {String id1=s.getString("email");
            String uname=s.getString("name");
            String uaddress=s.getString("alteid");
            jLabel4.setText("Email : "+id1+" Name is : "+uname+" Alternet
            EmailID is : "+uaddress+"\n"); }
        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(this,e.getMessage());
            e.printStackTrace();
        }

```

BUTTON—BACK

```

SearchContactForm loginform=new SearchContactForm();
        loginform.setVisible(true);
        setVisible(false);

```

• ***SEARCH PHONE NO.***

Events Inc.
Phone Number Details Form

Name

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    String id =jTextField1.getText();
```

```
String str="Select * from phone where name='"+id+"'";
```

```
ResultSet s=stmt.executeQuery(str);
```

```
while (s.next())
```

```
{
```

```
String id1=s.getString("email");
```

```
String uname=s.getString("name");
```

```
String uaddress=s.getString("phno");
```

```
jLabel4.setText(" Your Email : "+id1+" Name is : "+uname+" And  
Phone is : "+uaddress+"\n"); }
```

```
}
```

```
catch(Exception e)
```

```
{
```

```
JOptionPane.showMessageDialog(this,e.getMessage());
```

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

BUTTON—BACK

```
SearchContactForm loginform=new SearchContactForm();
```

```
loginform.setVisible(true);
```

```
setVisible(false);
```

• ***SEARCH BIRTHDAY***

Events Inc.
Search Birthday Form

Click Here

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    int id =jComboBox1.getSelectedIndex();
```

```

        id+=1;
String str="Select * from birthday where month="+id+"";
        ResultSet s=stmt.executeQuery(str);
        while (s.next())
        {
            String id1=s.getString("id");
            String uname=s.getString("name");
            String uaddress=s.getString("day");
            String ucity=s.getString("month");
            String ustate=s.getString("year");

            jLabel4.setText("ID : "+id1+" Name is : "+uname+" Day is : 
"+uaddress+" Month is : "+ ucity+" Year is : "+ustate+"\n");  }

        }
        catch(Exception e)
        {
            JOptionPane.showMessageDialog(this,e.getMessage());
            e.printStackTrace();}

```

BUTTON—BACK

```

SearchContactForm loginform=new SearchContactForm();
        loginform.setVisible(true);
        setVisible(false);

```

• ***SEARCH ANNIVERSARY***



Events Inc.
Search Anniversary Form

Click Here

Search **Back**

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    int id =jComboBox1.getSelectedIndex();
```



```

        id+=1;
String str="Select * from aniversry where month="+id+"";
ResultSet s=stmt.executeQuery(str);
        while (s.next())
        {
            String id1=s.getString("id");
            String uname=s.getString("name");
            String uaddress=s.getString("day");
            String ucity=s.getString("month");
            String ustate=s.getString("year");

            jLabel4.setText("ID : "+id1+" Name is : "+uname+" Day is : 
"+uaddress+" Month is : "+ ucity+" Year is : "+ustate+"\n");}

        }

        catch(Exception e)

        {JOptionPane.showMessageDialog(this,e.getMessage());
        e.printStackTrace();}

```

BUTTON—BACK

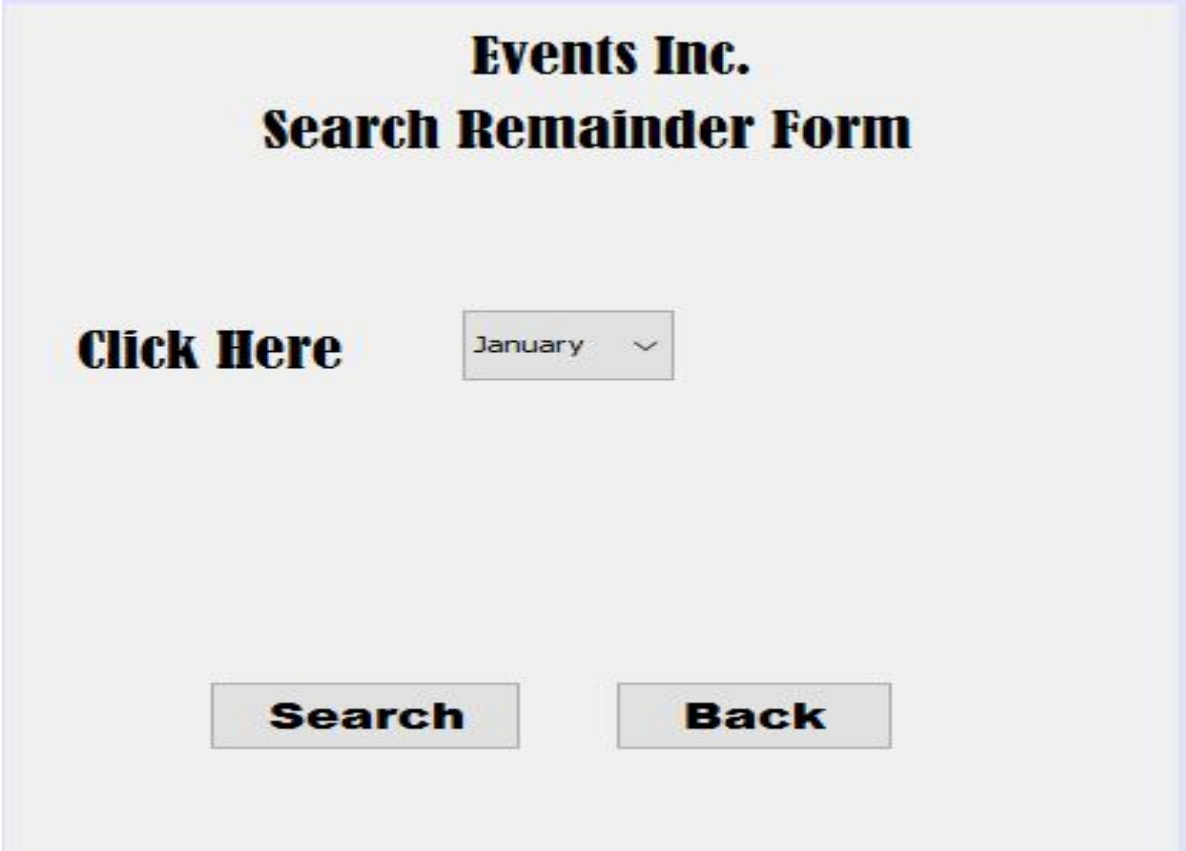
```

SearchContactForm loginform=new SearchContactForm();
        loginform.setVisible(true);

        setVisible(false);

```

• ***SEARCH REMAINDER***



Events Inc.
Search Remainder Form

Click Here

Search **Back**

CODING:

BUTTON-SEARCH

```
try
{
    Class.forName("com.mysql.jdbc.Driver");
    Connection con= (Connection)
DriverManager.getConnection("jdbc:mysql://localhost:3306/cms","root","");
    Statement stmt=con.createStatement();
    int id =jComboBox1.getSelectedIndex();
    id+=1;
```

```

String str="Select * from remainder where month="+id+"";
ResultSet s=stmt.executeQuery(str);
    while (s.next())
    {
        String id1=s.getString("id");
        String uname=s.getString("name");
        String uaddress=s.getString("rem_date");
        String uday=s.getString("day");
        String umonth=s.getString("month");
        String uyear=s.getString("year");

        jLabel4.setText(" ID is : "+id1+" Name is : "+uname+" Rem Date
is : "+uaddress+" Day is : "+uday+" Month is"+umonth+" year is :
        "+uyear+"\n");}
    }

    catch(Exception e)
    {JOptionPane.showMessageDialog(this,e.getMessage());
    e.printStackTrace();}

```

BUTTON—BACK

```

SearchContactForm loginform=new SearchContactForm();

    loginform.setVisible(true);

        setVisible(false);

```

D. BACK

```
MainEntryToTheProgramme loginform=new  
MainEntryToTheProgramme();  
loginform.setVisible(true);  
setVisible(false);
```

E. EXIT

```
System.exit(0);
```

STRUCTURE OF TABLES

The Tables Used in the making of this project are as follows:

1. ADDRESS TABLE

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(20)	YES		NULL	
address	varchar(40)	YES		NULL	
city	varchar(20)	YES		NULL	
state	varchar(20)	YES		NULL	
country	varchar(20)	YES		NULL	
pincode	int(11)	YES		NULL	

7 rows in set (0.01 sec)

2. EMAIL TABLE

Field	Type	Null	Key	Default	Extra
email	varchar(50)	YES		NULL	
name	varchar(20)	YES		NULL	
alteid	varchar(50)	YES		NULL	

3 rows in set (0.01 sec)

3. ANIVERSRY TABLE

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(20)	YES		NULL	
day	int(11)	YES		NULL	
month	int(11)	YES		NULL	
year	int(11)	YES		NULL	

5 rows in set (0.01 sec)

4. PHONE TABLE

Field	Type	Null	Key	Default	Extra
email	varchar(50)	YES		NULL	
name	varchar(20)	YES		NULL	
phno	int(11)	YES		NULL	

3 rows in set (0.01 sec)

5. BIRTHDAY TABLE

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(20)	YES		NULL	
day	int(11)	YES		NULL	
month	int(11)	YES		NULL	
year	int(11)	YES		NULL	

5 rows in set (0.01 sec)

6. REMAINDER TABLE

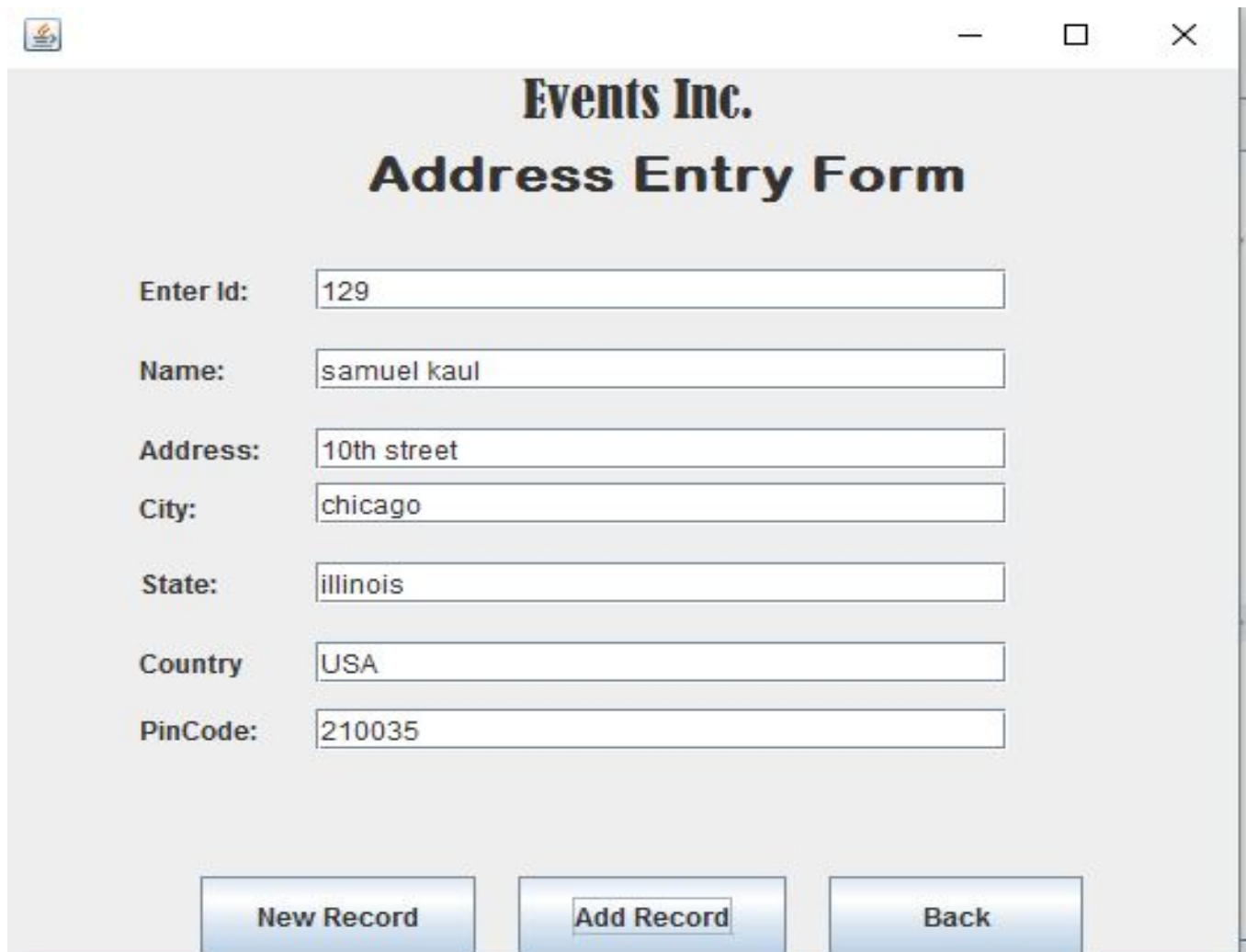
Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(20)	YES		NULL	
rem_date	int(11)	YES		NULL	
day	int(11)	YES		NULL	
month	int(11)	YES		NULL	
year	int(11)	YES		NULL	

6 rows in set (0.01 sec)

TEST REPORT

This report certifies that I have checked and run the project myself and it worked perfectly without any kind of problem.

The following are some results generated during the successful completion and run of the project:



The screenshot shows a web application window titled "Events Inc. Address Entry Form". The form contains several input fields with the following data: Enter Id: 129, Name: samuel kaul, Address: 10th street, City: chicago, State: illinois, Country: USA, and PinCode: 210035. At the bottom, there are three buttons: "New Record", "Add Record", and "Back".

Field	Value
Enter Id:	129
Name:	samuel kaul
Address:	10th street
City:	chicago
State:	illinois
Country	USA
PinCode:	210035

Buttons: New Record, Add Record, Back

When the add record button was clicked it stored the information in the database without any error and the same can be seen in the screenshot of the table given below:

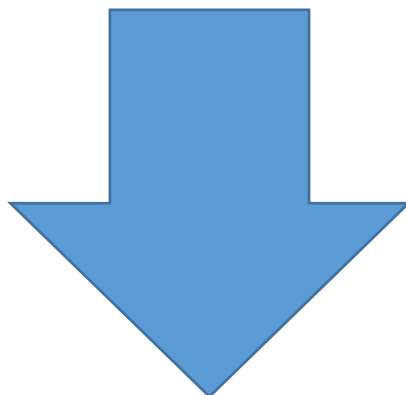
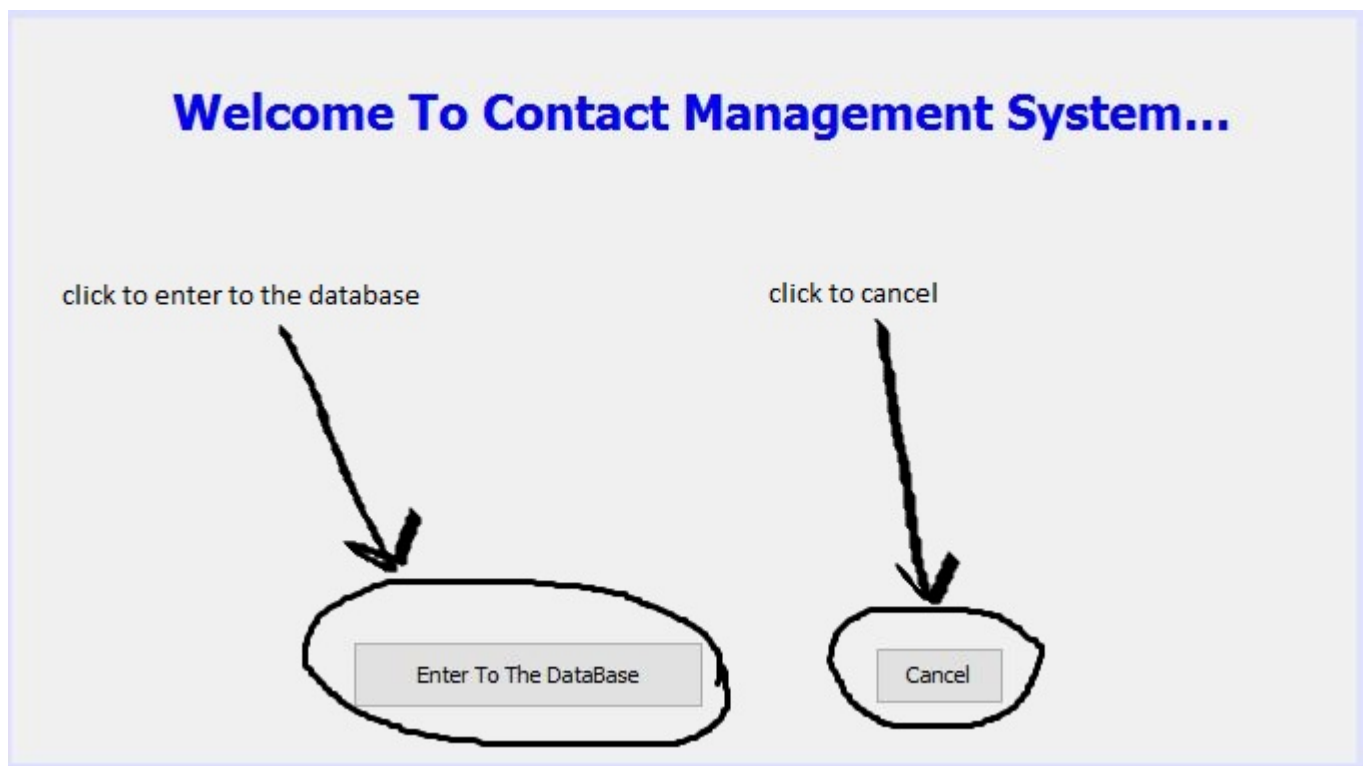
```
mysql> select * from address;
```

id	name	address	city	state	country	pincode
125	sabreesh bhatt	trikuta nagar	jammu	j&k	india	180012
126	abhay jamwal	gandhi nagar	jammu	j&k	india	180002
127	madhav sharma	trikuta nagar	jammu	j&k	india	180012
128	suvan tikku	trikuta nagar	jammu	j&k	india	180012
128	aneshwar sangra	panjtirthi	jammu	j&k	india	180012
129	samuel kaul	10th street	chicago	illinois	USA	210035

```
6 rows in set (0.00 sec)
```

USER/OPERATIONAL MANUAL

The following set of images are some instructions which will help in the successful and lag-free run of the project. Follow the guidelines given to run the project easily.



enter your login id

Login Id:

Password:

Login **Exit**

click to login click to exit

enter your password

A login form with two input fields, 'Login Id' and 'Password', each with a label and a text box. Below the 'Login Id' field is a 'Login' button with a label 'click to login' and an upward arrow. Below the 'Password' field is an 'Exit' button with a label 'click to exit' and an upward arrow. A label 'enter your password' with a rightward arrow points to the password input field. A large blue arrow points from the bottom of the login form to the top of the main menu.

click here to add contacts

Events Included!

Add Contacts

click here to update or delete contacts

Update/Delete Contact

click here to search for contacts

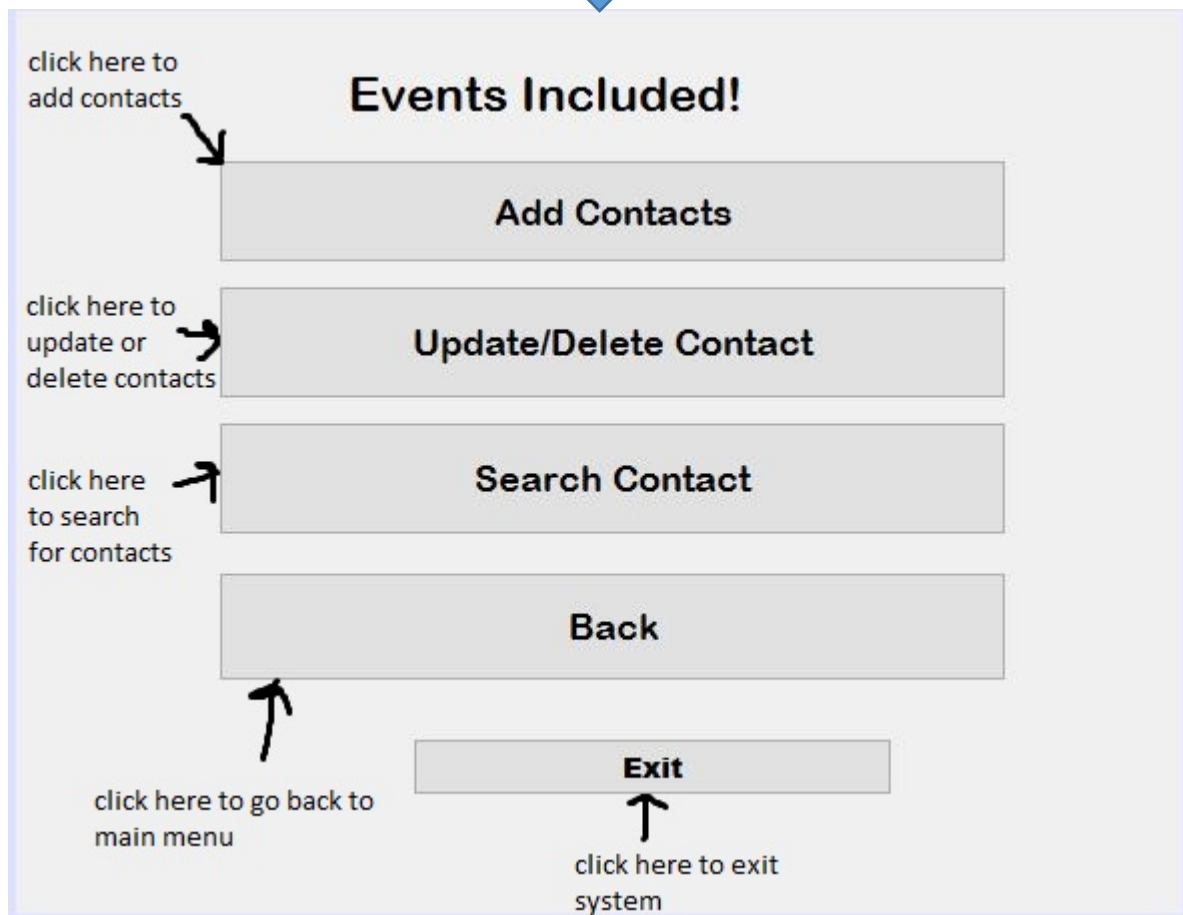
Search Contact

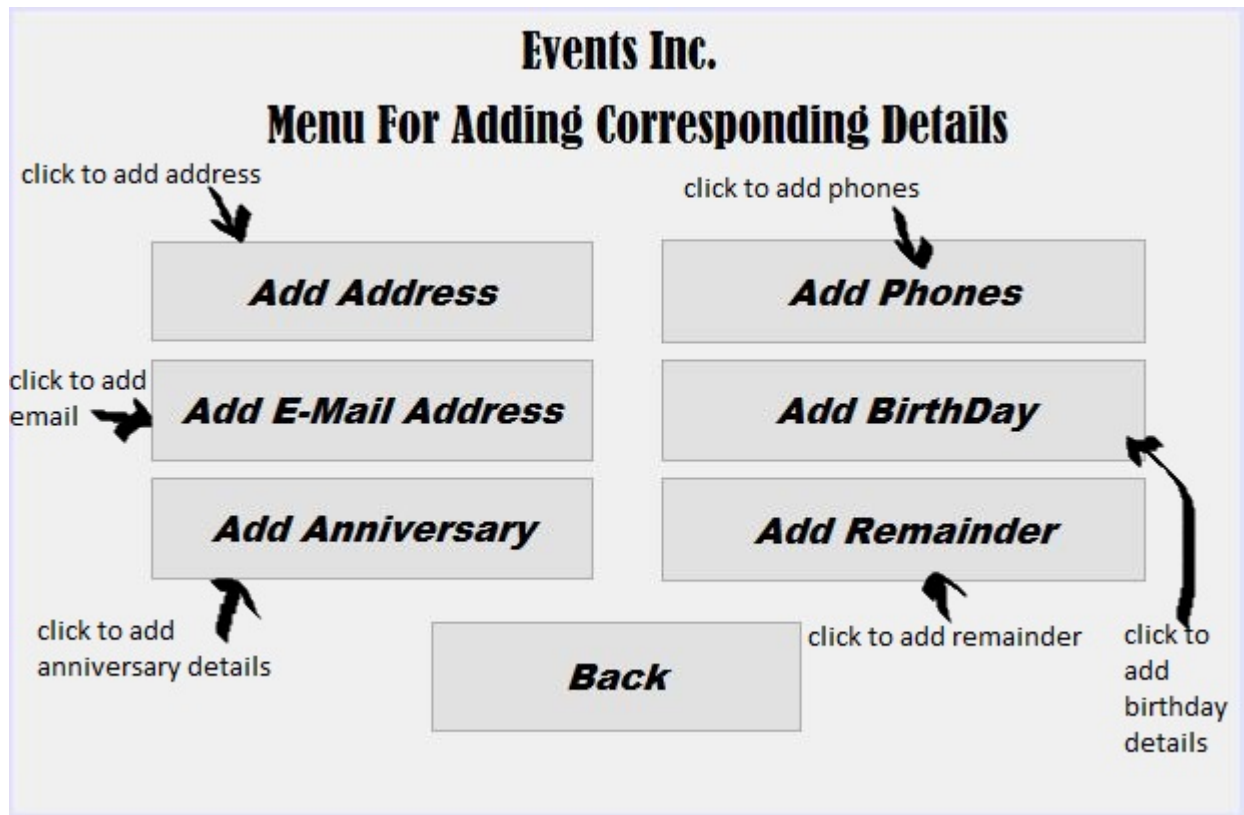
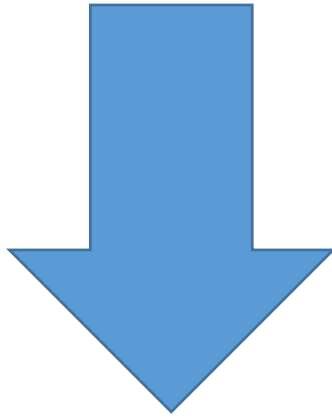
Back

click here to go back to main menu

Exit

click here to exit system

A main menu titled 'Events Included!' with four buttons: 'Add Contacts', 'Update/Delete Contact', 'Search Contact', and 'Back'. Each button has a label to its left with a rightward arrow pointing to it. Below the 'Back' button is an 'Exit' button with a label 'click here to exit system' and an upward arrow pointing to it. A label 'click here to go back to main menu' with an upward arrow points to the 'Back' button.



O R

Events Included!

Menu For Updating/Delete Corresponding Details

Update/Delete Address

click here to update or delete address

Update/Delete Phone No

click here to update or delete phone no

Update/Delete E-Mail

click here to update or delete email address

Update/Delete Birthday

click here to update or delete birthday details

Update/Delete Anniversary

click here to update or delete anniversary details

Update/Delete Remainder

click here to update or delete remainders

Back

click here to go back to previous window

OR

Events Inc.

Menu For Searching Corresponding Details

click here to search for address

Search Address

click here to search for emails

Search E-Mail

click here to search for phone no's

Search Phone.No

click here to search for birthday details

Search BirthDay

click here to search for anniversary details

Search Anniversary

click here to search for remainders

Search Remainder

Back

click here to go back to previous window

BRIEF BACKGROUND OF **THE ORGANISATION**

Broadways Inc., a multinational company headquartered in new delhi, india had the problem of storing the particulars of their employees. So they required a program which could solve their problem efficiently and could be user friendly so that each employee of their company can use and get benefited from it easily. So I came up with the idea of designing a contact management system which could the the required seamlessly and without causing any kind of trouble to its users. This program can solve their problem and can help them to maximum. In this manner they can manage the particulars very easily and can use them anytime and anywhere.

REFERENCE

- *INFORMATICS PRACTICES
TEXTBOOK FOR CLASS 12 BY NCERT
PUBLICATIONS*
- *INFORMATICS PRACTICES BY
SUMITA ARORA BY DHANPAT RAI
PUBLICATIONS*
- *ARIHANT'S ALL IN ONE FOR
INFORMATICS PRACTICES*
- www.meritnation.com
- www.studiestoday.com