

# **COGNITIVE DISORDERS AMONG KIDS**

A  
Mini Project Report

Submitted in partial fulfillment of the  
Requirements for the award of the Degree of

**BACHELOR OF ENGINEERING**

IN

**Department of Information Technology**

By

**DUGYALA VAMSHIDHAR RAO**

**1602-20-737-057**



**Department of Information Technology**  
**Vasavi College of Engineering (Autonomous)**  
**(Affiliated to Osmania University)**  
**Ibrahimbagh, Hyderabad-31**

**2022**

**Vasavi College of Engineering (Autonomous)  
(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



**DECLARATION BY THE CANDIDATE**

I, D.Vamshidhar Rao, hereby declare that the project report entitled **COGNITIVE DISORDERS AMONG KIDS** Department of Information Technology, VCE, Hyderabad, is submitted in partial fulfilment of the requirement for the award of the degree of **Bachelor of Engineering** in Information Technology.

This is a record of bonafide work carried out by me and the results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

**D.Vamshidharrao,  
1602-20-737-057.**

**Vasavi College of Engineering (Autonomous)**  
**(Affiliated to Osmania University)**

**Hyderabad-500 031**

**Department of Information Technology**



**BONAFIDE CERTIFICATE**

This is to certify that the project entitled **COGNITIVE DISORDERS AMONG KIDS** being submitted by **DUGYALA VAMSHIDHARRAO**, bearing **1602-20-737-057**, in partial fulfilment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science & Engineering is a record of bonafide work carried out by him/her under my guidance.

**DR. LEELAVATHY**

**Professor**

**Dept. of IT,**

## **ACKNOWLEDGEMENT**

With immense pleasure, we record our deep sense of gratitude to our guide Dr. Leelavathy, Professor, Vasavi College of Engineering, Hyderabad, for the valuable guidance and suggestions, keen interest and thorough encouragement extended throughout the period of the project work.

We express our thanks to all those who contributed for the successful completion of our project work.

# **TABLE OF CONTENTS:**

## **1. Abstract**

## **2. Introduction**

### **2.1 Overview**

### **2.2 Purpose**

### **2.4 Objective and Scope**

### **2.5 Project Description**

## **3.Design**

## **4. System Requirements**

### **4.1 Hardware Requirements**

### **4.2 Software Requirements**

## **5.Concepts Used**

## **6.Source Code**

## **7. Output/Result**

## **8. Conclusion**

## **9. References**

## **ABSTRACT:**

Cognitive disorders affect thinking and perceptual processes and the acquisition of knowledge and new information. They have an enormous societal impact because special educational resources are required, and independent living often cannot be achieved. Learning problems may lead to behavioral disorders in the home and community. The pathogenesis of most mild and moderate cognitive disorders is poorly understood. Severe cognitive impairment is usually accompanied by somatic abnormalities, and an etiology can be identified in many cases.

In this project, I created a user-friendly User interface that enables doctors or NGOs to store and analyse the cognitive diseases, their symptoms and children data suffering from those diseases.

## **INTRODUCTION:**

### **OVERVIEW:**

- This project provides a user- friendly UI that enables doctors or non-governmental organizations to store and analyse the cognitive diseases, their symptoms and children data suffering from those diseases

### **PURPOSE:**

- This project will be helpful in maintaining a database of children under the age of 14 suffering from different types of cognitive diseases and derive conclusions from the store data.
- This data can be used to analyse and find solutions to the conditions caused by this cognitive diseases and find out steps to decrease them and cured those diseases.

## DESIGN:

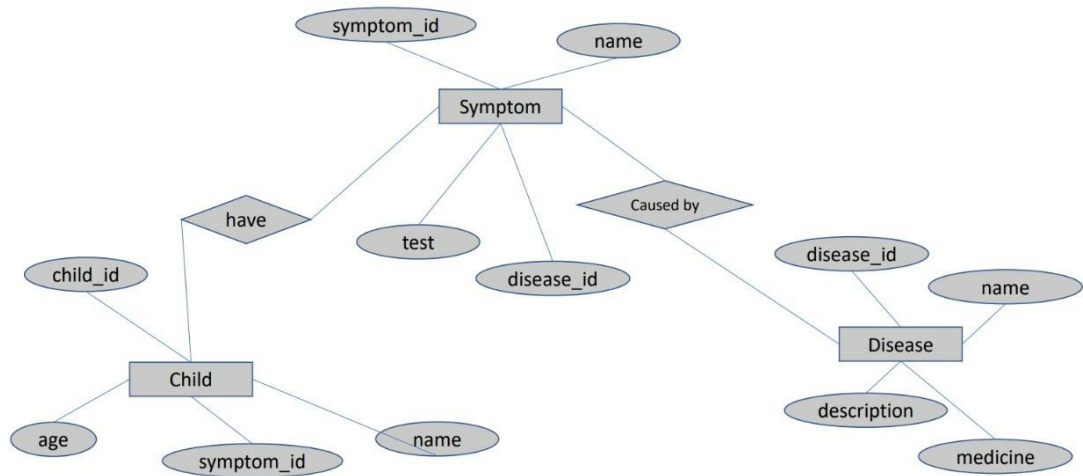
Considering the problem description given above, the problem can be solved as given here:

- We will be storing the data using 3 tables namely child, disease, symptoms.
- We will be storing the diseases with its description and prescribed medication.
- We will be also storing the symptoms and suggested tests to diagnose the disease causing that symptom.
- For storing the children data, while adding a child we give the child name, his or her age and the symptom the child is suffering from.
- Based on the symptom the database will automatically prescribe the test that needs to be taken to diagnose a disease that can be the reason for the symptom.
- It will also generate the medicine required to cure that disease and the whole data will be saved in a entry.

Synchronization of Barber thread and Customer threads



## E-R DIAGRAM:



## **SYSTEM REQUIREMENTS:**

### **HARDWARE REQUIREMENTS:**

- 1.Processor                    - core i5
2. RAM                        - 8 GB
- 3.Hard disk                  - 40GB
- 4.Mouse                      - Standard Mouse
- 5.Keyboard                  - Logitech Keyboard
6. Processor Speed   - 1.19GHz

### **SOFTWARE REQUIREMENTS:**

1. Operating System   - Microsoft Windows 10
2. Compiler – Java Compiler
3. MySQL Connector 29.0
4. MySQL Server 30.0

## CONCEPTS USED:

- DML operations
- JDBC connections
- Creating and maintaining tables
- ER Diagrams

## TABLE ANALYSIS:

### List of Tables:

- Child
- Symptom
- Disease

### List of attributes with their domain types:

#### Child

- Child\_id(number)
- name(varchar2)
- age(number)
- Symptom\_id(number)

#### Symptom

- Symptom\_id(number)
- name(varchar2)
- test(varchar2)
- disease\_id(number)

#### Disease

- disease\_id(number)
- name(varchar2)
- description(varchar2)
- medicine(number)

```

QL> create table Disease(
  2  effects VARCHAR2(100),
  3  disease_id NUMBER(5),
  4  disease_name VARCHAR2(20),
  5  cure VARCHAR2(20));

```

Table created.

```

QL> alter table Disease add primary key(disease_id);

```

Table altered.

```

QL> desc Disease

```

Name	Null?	Type
EFFECTS		VARCHAR2(100)
DISEASE_ID	NOT NULL	NUMBER(5)
DISEASE_NAME		VARCHAR2(20)
CURE		VARCHAR2(20)

```

QL>

```

```

SQL> create table Child(
  2  c_id NUMBER(10),
  3  c_name VARCHAR2(20),
  4  age NUMBER(3),
  5  disease_id NUMBER(5));

```

Table created.

```

SQL> alter table Child add primary key(c_id);

```

Table altered.

```

SQL> alter table Child add foreign key(disease_id) references Disease;

```

Table altered.

```

SQL> desc Child;

```

Name	Null?	Type
C_ID	NOT NULL	NUMBER(10)
C_NAME		VARCHAR2(20)
AGE		NUMBER(3)
DISEASE_ID		NUMBER(5)

```

SQL>

```

## FRONT END CONNECTION can be made as:

```
Public void connection {  
    Try{  
                                                Connection con =  
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","a  
mruth","vasavi");  
    }  
    catch (SQLException connectException){  
        System.out.println(connectException.getMessage());  
        System.out.println(connectException.getSQLState());  
        System.out.println(connectException.getErrorCode());System.exit(1);  
    }  
}
```

## SOURCE CODE:

➔ Menu.java file

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

public class Menu extends JFrame {
    private static final long serialVersionUID = 1L;
    static Menu frame;
    private JPanel contentPane;
    static String idd;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame = new Menu();
                    frame.setTitle("Cognitive Disorders among
Children");

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

```

        }
    }
});
}
public static void id(String id) {
    idd=id;
}
/**
 * Create the frame.
 */
public Menu() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 433);
    contentPane = new JPanel();
    contentPane.setForeground(Color.GRAY);
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);

    JLabel lblLibrarianSection = new JLabel("Cognitive Disorders
among Children");
    lblLibrarianSection.setFont(new Font("Academy Engraved LET",
Font.BOLD , 20));

    JButton btnBInfo = new JButton("Manage Children Data");
    btnBInfo.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            ManageChild.main(new String[]{});
            frame.dispose();
        }
    });
    btnBInfo.setFont(new Font("Tahoma", Font.PLAIN, 13));

    JButton btnSInfo = new JButton("Manage Diseases");
    btnSInfo.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent arg0) {
            ManageDiseases.main(new String[]{});

```

```

        frame.dispose();
    }
});
btnSInfo.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnChange = new JButton("Manage Symptoms");
btnChange.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ManageSymptoms.main(new String[]{});
        frame.dispose();
    }
});
btnChange.setFont(new Font("Tahoma", Font.PLAIN, 13));

GridLayout gl_contentPane = new GridLayout(contentPane);
gl_contentPane.setHorizontalGroup(

gl_contentPane.createParallelGroup(Alignment.LEADING)
    .addGroup(Alignment.TRAILING,
gl_contentPane.createSequentialGroup()
        .addContainerGap(81, Short.MAX_VALUE)
        .addComponent(lblLibrarianSection)
        .addGap(54))

    .addGroup(gl_contentPane.createSequentialGroup()
        .addGap(132)

    .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
        .addComponent(btnChange,
        GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
        .addComponent(btnSInfo,
        GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
        .addComponent(btnBInfo,
        GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE))
        .addContainerGap(101,
        Short.MAX_VALUE))

```



```

        );
        gl_contentPane.setVerticalGroup(

gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
            .addContainerGap()
            .addComponent(lblLibrarianSection)
            .addGap(18)
            .addComponent(btnBInfo,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)
            .addComponent(btnSInfo,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)

            .addComponent(btnChange,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)

        ));
        contentPane.setLayout(gl_contentPane);
    }

}

```

➔ Add child:

```

public class AddChild extends JFrame {
    private static final long serialVersionUID = 1L;
    static AddChild frame;
    private JPanel contentPane;
    private JTextField name;
    private JTextField age;
    private JTextField symptom;

    /**

```

```

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

/**
 * Create the frame.
 */
public AddChild() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 404);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);

    JLabel lblAddChild = new JLabel("Add Child");
    lblAddChild.setForeground(Color.GRAY);
    lblAddChild.setFont(new Font("Tahoma", Font.PLAIN, 18));

    JLabel lblName = new JLabel("Name :");

    JLabel lblAge = new JLabel("Age :");

    JLabel lblSymptom = new JLabel("Symptom :");

```

```

name = new JTextField();
name.setColumns(10);

age = new JTextField();
age.setColumns(10);

symptom = new JTextField();
symptom.setColumns(10);
JButton btnAddChild = new JButton("Submit");
btnAddChild.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String _name= name.getText();
        String _age= age.getText();
        String _symptom= symptom.getText();
        if(_name.equals("") || _age.equals("")||
        _symptom.equals("") ) {

            JOptionPane.showMessageDialog(AddChild.this,"Unknown Error !!!\n
            TextField Cannot be Blank");
        }
        else {
            if(SymptomDetails.checkSymptom(_name)) {

                JOptionPane.showMessageDialog(AddChild.this,"Book with Same
                Child ID is present already\nInsertion Failed !!!");
            }
            else {
                int i= ChildDetails.insertChild(_name, _age, _symptom);
                if(i>0){

                    JOptionPane.showMessageDialog(AddChild.this,"Child added
                    successfully!");
                }else{

                    JOptionPane.showMessageDialog(AddChild.this,"Unknown Error
                    !!!\nInsertion not completed");
                }
            }
        }
    }
});

```

```

        }}
    });

    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e1) {
            ManageChild.main(new String[]{});
            frame.dispose();
        }
    });

    GroupLayout gl_contentPane = new GroupLayout(contentPane);
    gl_contentPane.setHorizontalGroup(

gl_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl_contentPane.createSequentialGroup())

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(150)
                                .addComponent(lblAddChild))

.addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(18)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING,
false)
                                .addComponent(lblAge)

.addComponent(lblName)

.addComponent(lblSymptom)

                                )
                                .addGap(47)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

```

```

        .addComponent(symptom, GroupLayout.PREFERRED_SIZE, 167,
        GroupLayout.PREFERRED_SIZE)
                                .addComponent(age,
        GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
                                .addComponent(name,
        GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(125,
        Short.MAX_VALUE))
        .addGroup(Alignment.LEADING,
        gl_contentPane.createSequentialGroup()
        .addGap(50)
        .addComponent(btnAddChild)
        .addGap(30)
        .addComponent(btnBack)
        .addContainerGap())
        );
        gl_contentPane.setVerticalGroup(

        gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()
        .addComponent(lblAddChild)
        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
        .addComponent(lblName)
        .addComponent(name,
        GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
        .addComponent(lblAge)
        .addComponent(age,
        GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
        GroupLayout.PREFERRED_SIZE))
        .addGap(18)

```

```

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                    .addComponent(lblSymptom)
                    .addComponent(symptom,
 GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
 GroupLayout.PREFERRED_SIZE))
            .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                    )
            .addGap(30)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                    .addComponent(btnAddChild,
 GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

                    .addComponent(btnBack,
 GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE))

    )    );
    contentPane.setLayout(gl_contentPane);
}

}

```

Modify child:

```
public class ModifyChild extends JFrame {
```

```

    private static final long serialVersionUID = 1L;
    static ModifyChild frame;
    private JPanel contentPane;
    private JTextField childNameLbl;
    private JLabel childName;
    private JTextField ageLbl;
    private JLabel age;
    private JTextField symptomLbl;
    private JLabel symptom;

```

```
/**
```

```

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

```

```

/**
 * Create the frame.
 */
public ModifyChild() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(110, 95, 650, 350);
    setResizable(false);

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

    childNameLbl = new JTextField();
    childNameLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
    childNameLbl.setBounds(290, 37, 326, 40);
    contentPane.add(childNameLbl);
    childNameLbl.setColumns(10);

    ageLbl = new JTextField();

```

```

ageLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
ageLbl.setBounds(290, 87, 326, 40);
contentPane.add(ageLbl);
childNameLbl.setColumns(10);

symptomLbl = new JTextField();
symptomLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
symptomLbl.setBounds(290, 137, 326, 40);
contentPane.add(symptomLbl);
symptomLbl.setColumns(10);

childNameLbl.setText("");
    ageLbl.setText("");
    symptomLbl.setText("");

    Connection con = DB.getConnection();

    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.dispose();
            ManageChild.main(new String[]{});
        }
    });
    btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnBack.setBackground(new Color(240, 240, 240));
    btnBack.setBounds(450, 225, 130, 50);
    contentPane.add(btnBack);
    JButton btnSave = new JButton("Save");
    btnSave.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String child = childNameLbl.getText();
            String age = ageLbl.getText();
            String symptom = symptomLbl.getText();
            if(ChildDetails.checkChild(child)) {
                try {

```



```

        PreparedStatement st = con.prepareStatement("Update child set
age=?, symptom_id:= (select symptom_id from symptom where name=?
where name=?");
        st.setString(1, age);
        st.setString(2, symptom);
        st.setString(3, child);
        st.executeUpdate();
        System.out.println("updated info for : " + child);
        JOptionPane.showMessageDialog(btnSave, "Data Updated
Successfully :)");

    } catch (SQLException sqlException) {
        sqlException.printStackTrace();
    }

    else{

        JOptionPane.showMessageDialog(ModifyChild.this,"Child doesnot
exist !!!");

        childNameLbl.setText("");
        ageLbl.setText("");
        symptomLbl.setText("");

    }

}

});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnSave.setBackground(new Color(240, 240, 240));
btnSave.setBounds(300, 225, 130, 50);
contentPane.add(btnSave);

childName = new JLabel("Child name :");
childName.setFont(new Font("Tahoma", Font.PLAIN, 18));
childName.setBounds(45, 37, 326, 40);
contentPane.add(childName);

age = new JLabel("Age :");

```

```

age.setFont(new Font("Tahoma", Font.PLAIN, 18));
age.setBounds(45, 87, 326, 40);
contentPane.add(age);

```

```

symptom = new JLabel("Symptom :");
symptom.setFont(new Font("Tahoma", Font.PLAIN, 18));
symptom.setBounds(45, 137, 326, 40);
contentPane.add(symptom);

```

```

JButton btnLoad = new JButton("Load");
btnLoad.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String child = childNameLbl.getText();
        String title = ageLbl.getText();
        String author = symptomLbl.getText();
        Connection con = DB.getConnection();
        if(ChildDetails.checkChild(child)) {
            Statement st;
            try {
                System.out.println("load");
                PreparedStatement stmt = con.prepareStatement("select
child.name, age, symptom.name from child, symptom where child.name=?
and child.symptom_id=symptom.symptom_id");
                stmt.setString(1,child);
                ResultSet rs = stmt.executeQuery();
                while(rs.next())
                { childNameLbl.setText(rs.getString(1));
                  ageLbl.setText(rs.getString(2));
                  symptomLbl.setText(rs.getString(3));}
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }}else {

JOptionPane.showMessageDialog(ModifyChild.this,"Child is Invalid
!!!");

```

```

        childNameLbl.setText("");

```

```

        ageLbl.setText("");
        symptomLbl.setText("");
    }

    });
    btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnLoad.setBackground(new Color(240, 240, 240));
    btnLoad.setBounds(150, 225, 130, 50);
    contentPane.add(btnLoad);

}
}

```

View child:

```

public class ViewChild extends JFrame {

    private JPanel contentPane;
    private JTable table;
    static ViewChild frame = new ViewChild();

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setTitle("Children");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

/\*\*

\* Create the frame.

```

*/
public ViewChild() {
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);
    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Menu.main(new String[]{});
            frame.dispose();
        }
    });
    btnBack.setFont(new Font("Tahoma", Font.PLAIN, 13));

    String data[][]=null;
    String column[]=null;
    try{
        Connection con=DB.getConnection();
        PreparedStatement ps=con.prepareStatement("select child.name, age,
symptom.name , test, disease.name, descr, medicine from child, disease,
symptom where child.symptom_id=symptom.symptom_id and
symptom.disease_id=disease.disease_id
",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDATABLE);

        ResultSet rs=ps.executeQuery();
        ResultSetMetaData rsmd=rs.getMetaData();
        int cols=rsmd.getColumnCount();

        column = new String[]{"Name", "age", "symptom", "Test", "Disease",
"Description", "Medicine"};

        rs.last();
        int rows=rs.getRow();
        rs.beforeFirst();
    }
}

```

```

        data=new String[rows][cols];
        int count=0;
        while(rs.next()){
            for(int i=1;i<=cols;i++){
                data[count][i-1]=rs.getString(i);
            }
            count++;
        }
        con.close();
    }catch(Exception e){System.out.println(e);}

    table = new JTable(data,column);
    table.setDefaultEditor(Object.class, null);
    JScrollPane sp=new JScrollPane(table);

    contentPane.add(sp, BorderLayout.CENTER);
}

}

Child details:
public class ChildDetails {

    public static boolean checkChild(String childName){
        boolean status=false;
        try{
            Connection con=DB.getConnection();
            PreparedStatement ps=con.prepareStatement("select *
from child where name=?");
            ps.setString(1,childName);
            ResultSet rs=ps.executeQuery();
            status=rs.next();

            con.close();
        }catch(Exception e){System.out.println(e);}
        return status;
    }
}

```

```

        public static int insertChild(String childName, String age, String
symptom){
            int status=0;
            try{
                Connection con=DB.getConnection();
                PreparedStatement ps=con.prepareStatement("insert into
child(name,age,symptom_id) values(?,?,(select symptom_id from symptom
where name=?))");
                ps.setString(1,childName);
                ps.setString(2,age);
                ps.setString(3,symptom);
                status=ps.executeUpdate();
                con.close();
            }catch(Exception e){System.out.println(e);}
            return status;
        }

```

```

        public static int deleteChild(String name){
            int status=0;
            try{
                Connection con=DB.getConnection();
                PreparedStatement ps=con.prepareStatement("delete
from child where name=?");
                ps.setString(1,name);
                status=ps.executeUpdate();
                con.close();
            }catch(Exception e){System.out.println(e);}
            return status;
        }
    }

```

Delete Child:

```

public class DeleteChild extends JFrame {
    static DeleteChild frame;
    private JPanel contentPane;
    private JTextField child;

```

```

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

/**
 * Create the frame.
 */
public DeleteChild() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 200);

    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);

    JLabel lblEnterName = new JLabel("Enter Child name :");

    child = new JTextField();
    child.setColumns(10);

    JButton btnDelete = new JButton("Delete");
    btnDelete.addActionListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent e) {
            String childName= child.getText();
            if(childName==null||childName.trim().equals("")){

                JOptionPane.showMessageDialog(DeleteChild.this,"Child name can't
                be blank");

            }else{

                int i=ChildDetails.deleteChild(childName);
                if(i>0){

                    JOptionPane.showMessageDialog(DeleteChild.this,"Child entry deleted
                    successfully!");

                }else{

                    JOptionPane.showMessageDialog(DeleteChild.this,"Unable to delete
                    given child");

                }

            }

        }

    });
    btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));

    JButton btnNewButton = new JButton("Back");
    btnNewButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            ManageChild.main(new String[]{});
            frame.dispose();
        }
    });
    btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 13));
    GroupLayout gl_contentPane = new GroupLayout(contentPane);
    gl_contentPane.setHorizontalGroup(

        gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()

            .addGap(39)

```



```

                                .addComponent(lblEnterName)
                                .addGap(57)
                                .addComponent(child,
GroupLayout.PREFERRED_SIZE, 178, GroupLayout.PREFERRED_SIZE)
                                .addContainerGap(107,
Short.MAX_VALUE))

        .addGroup(gl_contentPane.createSequentialGroup())
                                .addGap(100)
                                .addComponent(btnDelete)
                                .addGap(30)
                                .addComponent(btnNewButton)
                                )
        );
        gl_contentPane.setVerticalGroup(
            gl_contentPane.createParallelGroup(Alignment.CENTER)

        .addGroup(gl_contentPane.createSequentialGroup())
                                .addGap(19)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                .addComponent(child,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
                                .addComponent(lblEnterName))
                                .addGap(33)

        .addGroup(gl_contentPane.createParallelGroup())
                                .addComponent(btnDelete)

        .addComponent(btnNewButton)
        ));
        contentPane.setLayout(gl_contentPane);
    }
}

```

SYMPTOM UI:

ADD symptom:

```
public class AddSymptom extends JFrame {
```

```

private static final long serialVersionUID = 1L;
static AddSymptom frame;
private JPanel contentPane;
private JTextField name;
private JTextField description;
private JTextField disease;

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

/**
 * Create the frame.
 */
public AddSymptom() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 404);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);

    JLabel lblAddSymptom = new JLabel("Add Symptom");
    lblAddSymptom.setForeground(Color.GRAY);

```

```

lblAddSymptom.setFont(new Font("Tahoma", Font.PLAIN, 18));

JLabel lblName = new JLabel("Symptom Name :");

JLabel lblDescription = new JLabel("Description :");

JLabel lblDisease = new JLabel("Disease :");

name = new JTextField();
name.setColumns(10);

description = new JTextField();
description.setColumns(10);

disease = new JTextField();
disease.setColumns(10);
JButton btnAddChild = new JButton("Submit");
btnAddChild.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String _name= name.getText();
        String _description= description.getText();
        String _disease= disease.getText();
        if(_name.equals("") || _description.equals("")||
        _disease.equals("")) {

            JOptionPane.showMessageDialog(AddSymptom.this,"Unknown Error
            !!!\n TextField Cannot be Blank");
        }
        else {
            if(SymptomDetails.checkSymptom(_name)) {

                JOptionPane.showMessageDialog(AddSymptom.this,"Symptom with
                Same Symptom Name is present already\nInsertion Failed !!!");
            }
            else {
                int i= SymptomDetails.insertSymptom(_name,
                _description, _disease);
            }
        }
    }
});

```

```

        if(i>0){

            JOptionPane.showMessageDialog(AddSymptom.this,"Symptom added
            successfully!");

        }else{

            JOptionPane.showMessageDialog(AddSymptom.this,"Unknown Error
            !!!\nInsertion not completed");

        }
    }
}

});

JButton btnBack = new JButton("Back");
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e1) {
        ManageSymptoms.main(new String[]{});
        frame.dispose();
    }
});

GroupLayout gl_contentPane = new GroupLayout(contentPane);
gl_contentPane.setHorizontalGroup(

gl_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl_contentPane.createSequentialGroup()

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(150)

.addComponent(lblAddSymptom))

.addGroup(gl_contentPane.createSequentialGroup()
                                .addGap(18)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING,
false)

```

```

        .addComponent(lblDescription)

        .addComponent(lblName)

        .addComponent(lblDisease)

        )

        .addGap(47)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

                .addComponent(disease, GroupLayout.PREFERRED_SIZE, 167,
                GroupLayout.PREFERRED_SIZE)

                .addComponent(description, GroupLayout.PREFERRED_SIZE, 167,
                GroupLayout.PREFERRED_SIZE)

                .addComponent(name,
                GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))

        .addContainerGap(125,
        Short.MAX_VALUE))

        .addGroup(Alignment.LEADING,
        gl_contentPane.createSequentialGroup()

                .addGap(50)

                .addComponent(btnAddChild)

                .addGap(30)

                .addComponent(btnBack)

                .addContainerGap())

        );

        gl_contentPane.setVerticalGroup(

        gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()

                .addComponent(lblAddSymptom)

                .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)

```

```

                                .addComponent(lblName)
                                .addComponent(name,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

                                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                .addComponent(lblDescription)
                                .addComponent(description,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

                                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                .addComponent(lblDisease)
                                .addComponent(disease,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

                                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                )
                                .addGap(30)

                                .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                .addComponent(btnAddChild,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

                                .addComponent(btnBack,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE))

                                )    );
                                contentPane.setLayout(gl_contentPane);
                                }

}

```

Delete symptom:

```

public class DeleteSymptom extends JFrame {
    static DeleteSymptom frame;

```

```

private JPanel contentPane;
private JTextField symptom;

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

/**
 * Create the frame.
 */
public DeleteSymptom() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 450, 200);

    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

    setContentPane(contentPane);

    JLabel lblEnterName = new JLabel("Enter Symptom name :");

    symptom = new JTextField();
    symptom.setColumns(10);

```

```

        JButton btnDelete = new JButton("Delete");
        btnDelete.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String symptomName= symptom.getText();

                if(symptomName==null||symptomName.trim().equals("")){

                    JOptionPane.showMessageDialog(DeleteSymptom.this,"Symptom
name can't be blank");
                }else{

                    int
i=SymptomDetails.deleteSymptom(symptomName);
                    if(i>0){

                        JOptionPane.showMessageDialog(DeleteSymptom.this,"Symptom
entry deleted successfully!");
                    }else{

                        JOptionPane.showMessageDialog(DeleteSymptom.this,"Unable to
delete given symptom");
                    }
                }
            }
        });
        btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));

        JButton btnNewButton = new JButton("Back");
        btnNewButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                ManageDiseases.main(new String[]{});
                frame.dispose();
            }
        });
        btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 13));
        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(

```



```

gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
    .addGap(39)
    .addComponent(lblEnterName)
    .addGap(57)
    .addComponent(symptom,
 GroupLayout.PREFERRED_SIZE, 178, GroupLayout.PREFERRED_SIZE)
    .addContainerGap(107,
 Short.MAX_VALUE))

.addGroup(gl_contentPane.createSequentialGroup()
    .addGap(100)
    .addComponent(btnDelete)
    .addGap(30)
    .addComponent(btnNewButton)
    )
);
gl_contentPane.setVerticalGroup(
    gl_contentPane.createParallelGroup(Alignment.CENTER)

.addGroup(gl_contentPane.createSequentialGroup()
    .addGap(19)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
    .addComponent(symptom,
 GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
 GroupLayout.PREFERRED_SIZE)
    .addComponent(lblEnterName))
    .addGap(33)

.addGroup(gl_contentPane.createParallelGroup()
    .addComponent(btnDelete)

.addComponent(btnNewButton)
    ));
contentPane.setLayout(gl_contentPane);

```

```

    }
}

```

Manage Symptom:

```

public class ManageSymptoms extends JFrame {
    private static final long serialVersionUID = 1L;
    static ManageSymptoms frame;
    private JPanel contentPane;

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

    /**
     * Create the frame.
     */
    public ManageSymptoms() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 433);
        contentPane = new JPanel();
        contentPane.setForeground(Color.GRAY);
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);

        JLabel lblSection = new JLabel("Manage Symptoms");
    }
}

```

```

lblSection.setFont(new Font("Tahoma", Font.PLAIN, 22));

JButton btnAdd = new JButton("Add Symptom");
btnAdd.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        AddSymptom.main(new String[]{});
        frame.dispose();
    }
});
btnAdd.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnSUpdate = new JButton("Modify Symptom details");
btnSUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        ModifySymptoms.main(new String[]{});
        frame.dispose();
    }
});
btnSUpdate.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnSDelete = new JButton("Delete Symptom");
btnSDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        DeleteSymptom.main(new String[]{});
        frame.dispose();
    }
});
btnSDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnView = new JButton("View All Symptoms");
btnView.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ViewSymptoms.main(new String[]{});
    }
});

```

```

        btnView.setFont(new Font("Tahoma", Font.PLAIN, 13));

        JButton btnBack = new JButton("Back");
        btnBack.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Menu.main(new String[]{});
                frame.dispose();
            }
        });
        btnBack.setFont(new Font("Tahoma", Font.PLAIN, 13));

        GroupLayout gl_contentPane = new GroupLayout(contentPane);
        gl_contentPane.setHorizontalGroup(

            gl_contentPane.createParallelGroup(Alignment.LEADING)
                .addGroup(Alignment.TRAILING,
gl_contentPane.createSequentialGroup()
                    .addContainerGap(81, Short.MAX_VALUE)
                    .addComponent(lblSection)
                    .addGap(54))

            .addGroup(gl_contentPane.createSequentialGroup()
                .addGap(132)

            .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                .addComponent(btnBack,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
                .addComponent(btnView,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
                .addComponent(btnSDelete,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
                .addComponent(btnSUpdate,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
                .addComponent(btnAdd,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE))
                .addContainerGap(101,
Short.MAX_VALUE))
        );

```

```

        gl_contentPane.setVerticalGroup(

gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
            .addContainerGap()
            .addComponent(lblSection)
            .addGap(18)
            .addComponent(btnAdd,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)
            .addComponent(btnSUpdate,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)
            .addComponent(btnSDelete,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)
            .addComponent(btnView,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addGap(18)
            .addComponent(btnBack,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
            .addContainerGap(16, Short.MAX_VALUE))
        );
        contentPane.setLayout(gl_contentPane);
    }

}

```

Modify symptom:

```
public class ModifySymptoms extends JFrame {
```

```

    private static final long serialVersionUID = 1L;
    static ModifySymptoms frame;
    private JPanel contentPane;
    private JTextField symptomNameLbl;
    private JLabel symptomsName;
    private JTextField diseaseLbl;
    private JLabel disease;

```

```

private JTextField testsLbl;
private JLabel tests;

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

/**
 * Create the frame.
 */
public ModifySymptoms() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(110, 95, 650, 350);
    setResizable(false);

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

    symptomNameLbl = new JTextField();
    symptomNameLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
    symptomNameLbl.setBounds(290, 37, 326, 40);

```

```

contentPane.add(symptomNameLbl);
symptomNameLbl.setColumns(10);

diseaseLbl = new JTextField();
diseaseLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
diseaseLbl.setBounds(290, 87, 326, 40);
contentPane.add(diseaseLbl);
symptomNameLbl.setColumns(10);

testsLbl = new JTextField();
testsLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
testsLbl.setBounds(290, 137, 326, 40);
contentPane.add(testsLbl);
testsLbl.setColumns(10);

symptomNameLbl.setText("");
diseaseLbl.setText("");
testsLbl.setText("");

Connection con = DB.getConnection();

JButton btnBack = new JButton("Back");
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        frame.dispose();
        ManageSymptoms.main(new String[]{});
    }
});
btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnBack.setBackground(new Color(240, 240, 240));
btnBack.setBounds(450, 225, 130, 50);
contentPane.add(btnBack);

JButton btnSave = new JButton("Save");
btnSave.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String b = symptomNameLbl.getText();
        String t = diseaseLbl.getText();

```

```

        String a = testsLbl.getText();
        if(SymptomDetails.checkSymptom(b)) {
            try {
                PreparedStatement st = con.prepareStatement("Update symptom
set disease_id:= (select disease_id from disease where name=?), test= ?
where name=?");
                st.setString(1, t);
                st.setString(2, a);
                st.setString(3, b);
                st.executeUpdate();
                System.out.println("updated info for : " + b);
                JOptionPane.showMessageDialog(btnSave, "Data Updated
Successfully :)");

            } catch (SQLException sqlException) {
                sqlException.printStackTrace();
            }
        }

        else{

            JOptionPane.showMessageDialog(ModifySymptoms.this,"Symptom
doesnot exist !!!");

            symptomNameLbl.setText("");
            diseaseLbl.setText("");
            testsLbl.setText("");

        }

    }

});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnSave.setBackground(new Color(240, 240, 240));
btnSave.setBounds(300, 225, 130, 50);
contentPane.add(btnSave);

symptomsName = new JLabel("Symptom name :");
symptomsName.setFont(new Font("Tahoma", Font.PLAIN, 18));
symptomsName.setBounds(45, 37, 326, 40);

```



```

contentPane.add(symptomsName);

disease = new JLabel("Suspected Disease :");
disease.setFont(new Font("Tahoma", Font.PLAIN, 18));
disease.setBounds(45, 87, 326, 40);
contentPane.add(disease);

tests = new JLabel("Suggested Test :");
tests.setFont(new Font("Tahoma", Font.PLAIN, 18));
tests.setBounds(45, 137, 326, 40);
contentPane.add(tests);

String test = symptomNameLbl.getText();
symptomNameLbl.addKeyListener((KeyListener) new KeyAdapter() {
    @Override
    public void keyPressed(KeyEvent e) {
        if(e.getKeyCode() == KeyEvent.VK_ENTER){
            String b = symptomNameLbl.getText();
            String t = diseaseLbl.getText();
            String a = testsLbl.getText();
            Connection con = DB.getConnection();
            if(SymptomDetails.checkSymptom(b)) {
                Statement st;
                try {
                    PreparedStatement stmt =
con.prepareStatement("select name, disease, test from symptom, disease
where symptom.name=? and symptom.disease_id=disease.disease_id");
                    stmt.setString(1,b);
                    ResultSet rs = stmt.executeQuery();
                    while(rs.next())
                    {
                        symptomNameLbl.setText(rs.getString(2));
                        diseaseLbl.setText(rs.getString(3));
                        testsLbl.setText(rs.getString(4));}
                    } catch (SQLException e1) {
                        e1.printStackTrace();
                    }
                }
            }
        }
    }
});

```

```

        }else {

            JOptionPane.showMessageDialog(ModifySymptoms.this,"Symptom is
not listed !!!");

            symptomNameLbl.setText("");
            diseaseLbl.setText("");
            testsLbl.setText("");
        }

    }

    });

    JButton btnLoad = new JButton("Load");
    btnLoad.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String book = symptomNameLbl.getText();
            String title = diseaseLbl.getText();
            String author = testsLbl.getText();
            Connection con = DB.getConnection();
            if(SymptomDetails.checkSymptom(book)) {
                Statement st;
                try {
                    PreparedStatement stmt = con.prepareStatement("select
symptom.name, disease.name, test from symptom, disease where
symptom.name=? and symptom.disease_id=disease.disease_id");
                    stmt.setString(1,book);
                    ResultSet rs = stmt.executeQuery();
                    while(rs.next())
                    {symptomNameLbl.setText(rs.getString(1));
                     diseaseLbl.setText(rs.getString(2));
                     testsLbl.setText(rs.getString(3));}
                } catch (SQLException e1) {
                    e1.printStackTrace();
                }}else {

```

```
JOptionPane.showMessageDialog(ModifySymptoms.this,"Book ID is  
Invalid !!!");
```

```
    symptomNameLbl.setText("");  
    diseaseLbl.setText("");  
    testsLbl.setText("");
```

```
    }
```

```
    });
```

```
    btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
```

```
    btnLoad.setBackground(new Color(240, 240, 240));
```

```
    btnLoad.setBounds(150, 225, 130, 50);
```

```
    contentPane.add(btnLoad);
```

```
    }
```

```
}
```

Symptom Details:

```
public class SymptomDetails {
```

```
    public static boolean checkSymptom(String SymptomName){
```

```
        boolean status=false;
```

```
        try{
```

```
            Connection con=DB.getConnection();
```

```
            PreparedStatement ps=con.prepareStatement("select *  
from symptom where name=?");
```

```
            ps.setString(1,SymptomName);
```

```
            ResultSet rs=ps.executeQuery();
```

```
            status=rs.next();
```

```
            con.close();
```

```
        }catch(Exception e){System.out.println(e);} 
```

```
        return status;
```

```
    }
```

```
    public static int insertSymptom(String symptomName, String  
description, String disease){
```

```
        int status=0;
```

```
        try{
```

```

        Connection con=DB.getConnection();
        PreparedStatement ps=con.prepareStatement("insert into
symptom(name,test,disease_id) values(?,?,(select disease_id from disease
where name=?))");
        ps.setString(1,symptomName);
        ps.setString(2,description);
        ps.setString(3,disease);
        status=ps.executeUpdate();
        con.close();
    }catch(Exception e){System.out.println(e);}
    return status;
}

public static int deleteSymptom(String name){
    int status=0;
    try{
        Connection con=DB.getConnection();
        PreparedStatement ps=con.prepareStatement("delete
from symptom where name=?");
        ps.setString(1,name);
        status=ps.executeUpdate();
        con.close();
    }catch(Exception e){System.out.println(e);}
    return status;
}
}

```

View Symptom:

```

public class ViewSymptoms extends JFrame {

    private JPanel contentPane;
    private JTable table;
    static ViewSymptoms frame = new ViewSymptoms();

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

```

```

        try {
            frame.setTitle("Symptoms");
            frame.setVisible(true);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

});
}

/**
 * Create the frame.
 */
public ViewSymptoms() {
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);
    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Menu.main(new String[]{});
            frame.dispose();
        }
    });
    btnBack.setFont(new Font("Tahoma", Font.PLAIN, 13));

    String data[][]=null;
    String column[]=null;
    try{
        Connection con=DB.getConnection();
        PreparedStatement ps=con.prepareStatement("select symptom_id,
symptom.name, disease.name, test from symptom, disease where

```

```
symptom.disease_id=disease.disease_id",ResultSet.TYPE_SCROLL_SENSI
TIVE,ResultSet.CONCUR_UPDATABLE);
```

```
    ResultSet rs=ps.executeQuery();
```

```
    ResultSetMetaData rsmd=rs.getMetaData();
```

```
    int cols=rsmd.getColumnCount();
```

```
    column=new String[cols];
```

```
    for(int i=1;i<=cols;i++){
```

```
        column[i-1]=rsmd.getColumnName(i);
```

```
    }
```

```
    rs.last();
```

```
    int rows=rs.getRow();
```

```
    rs.beforeFirst();
```

```
    data=new String[rows][cols];
```

```
    int count=0;
```

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

```
        for(int i=1;i<=cols;i++){
```

```
            data[count][i-1]=rs.getString(i);
```

```
        }
```

```
        count++;
```

```
    }
```

```
    con.close();
```

```
}catch(Exception e){System.out.println(e);}


```

```
table = new JTable(data,column);
```

```
table.setDefaultEditor(Object.class, null);
```

```
JScrollPane sp=new JScrollPane(table);
```

```
contentPane.add(sp, BorderLayout.CENTER);
```

```
}
```

```
}
```

Disease UI:

ADD disease:

```

public class AddDisease extends JFrame {
    private static final long serialVersionUID = 1L;
    static AddDisease frame;
    private JPanel contentPane;
    private JTextField name;
    private JTextField description;
    private JTextField medication;

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

    /**
     * Create the frame.
     */
    public AddDisease() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 404);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);

        JLabel lblAddDisease = new JLabel("Add Disease");
    }
}

```

```

lblAddDisease.setForeground(Color.GRAY);
lblAddDisease.setFont(new Font("Tahoma", Font.PLAIN, 18));

JLabel lblName = new JLabel("Disease Name :");

JLabel lblDescription = new JLabel("Description :");

JLabel lblMedication = new JLabel("Prescribed medicine :");

name = new JTextField();
name.setColumns(10);

description = new JTextField();
description.setColumns(10);

medication = new JTextField();
medication.setColumns(10);
JButton btnAddChild = new JButton("Submit");
btnAddChild.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String _name= name.getText();
        String _description= description.getText();
        String _medication= medication.getText();
        if(_name.equals("") || _description.equals("")||
        _medication.equals("") ) {

            JOptionPane.showMessageDialog(AddDisease.this,"Unknown Error
            !!!\n TextField Cannot be Blank");
        }
        else {
            if(SymptomDetails.checkSymptom(_name)) {

                JOptionPane.showMessageDialog(AddDisease.this,"Disease with
                Same Disease Name is present already\nInsertion Failed !!!");
            }
            else {

```



```

        int i= DiseaseDetails.insertDisease(_name, _description,
        _medication);

        if(i>0){

            JOptionPane.showMessageDialog(AddDisease.this,"Disease added
            successfully!");

        }else{

            JOptionPane.showMessageDialog(AddDisease.this,"Unknown Error
            !!!\nInsertion not completed");

        }
        }
    }}

});

JButton btnBack = new JButton("Back");
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e1) {
        ManageDiseases.main(new String[]{});
        frame.dispose();
    }
});

GroupLayout gl_contentPane = new GroupLayout(contentPane);
gl_contentPane.setHorizontalGroup(

gl_contentPane.createParallelGroup(Alignment.TRAILING)

.addGroup(gl_contentPane.createSequentialGroup()

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
                .addGap(150)

.addComponent(lblAddDisease))

.addGroup(gl_contentPane.createSequentialGroup()
                .addGap(18)

```

```

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING,
false)

        .addComponent(lblDescription)

        .addComponent(lblName)

        .addComponent(lblMedication)

        )
        .addGap(47)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addComponent(medication, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)

        .addComponent(description, GroupLayout.PREFERRED_SIZE, 167,
GroupLayout.PREFERRED_SIZE)

        .addComponent(name,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))
        .addContainerGap(125,
Short.MAX_VALUE))

        .addGroup(Alignment.LEADING,
gl_contentPane.createSequentialGroup()

        .addGap(50)
        .addComponent(btnAddChild)
        .addGap(30)
        .addComponent(btnBack)
        .addContainerGap())

    );
    gl_contentPane.setVerticalGroup(

gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()

        .addComponent(lblAddDisease)

```

```

        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
            .addComponent(lblName)
            .addComponent(name,
                GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
                GroupLayout.PREFERRED_SIZE))
        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
            .addComponent(lblDescription)
            .addComponent(description,
                GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
                GroupLayout.PREFERRED_SIZE))
        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
            .addComponent(lblMedication)
            .addComponent(medication,
                GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
                GroupLayout.PREFERRED_SIZE))
        .addGap(18)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
            )
        .addGap(30)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
            .addComponent(btnAddChild,
                GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

            .addComponent(btnBack,
                GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE))

        )    );
        contentPane.setLayout(gl_contentPane);
    }

}

```

Delete disease:

```
public class DeleteDisease extends JFrame {
    static DeleteDisease frame;
    private JPanel contentPane;
    private JTextField disease;

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

    /**
     * Create the frame.
     */
    public DeleteDisease() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 200);

        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));

        setContentPane(contentPane);
    }
}
```

```

JLabel lblEnterName = new JLabel("Enter Disease name :");

disease = new JTextField();
disease.setColumns(10);

JButton btnDelete = new JButton("Delete");
btnDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String diseaseName= disease.getText();

        if(diseaseName==null||diseaseName.trim().equals("")){

            JOptionPane.showMessageDialog(DeleteDisease.this,"Disease name
            can't be blank");

        }else{

            int
            i=DiseaseDetails.deleteDisease(diseaseName);
            if(i>0){

                JOptionPane.showMessageDialog(DeleteDisease.this,"Disease entry
                deleted successfully!");

            }else{

                JOptionPane.showMessageDialog(DeleteDisease.this,"Unable to
                delete given disease");

            }

        }

    }
});
btnDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnNewButton = new JButton("Back");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        ManageDiseases.main(new String[]{});
        frame.dispose();
    }
}

```

```

});
btnNewButton.setFont(new Font("Tahoma", Font.PLAIN, 13));
GroupLayout gl_contentPane = new GroupLayout(contentPane);
gl_contentPane.setHorizontalGroup(

gl_contentPane.createParallelGroup(Alignment.LEADING)

.addGroup(gl_contentPane.createSequentialGroup()
            .addGap(39)
            .addComponent(lblEnterName)
            .addGap(57)
            .addComponent(disease,
GroupLayout.PREFERRED_SIZE, 178, GroupLayout.PREFERRED_SIZE)
            .addContainerGap(107,
Short.MAX_VALUE))

.addGroup(gl_contentPane.createSequentialGroup()
            .addGap(100)
            .addComponent(btnDelete)
            .addGap(30)
            .addComponent(btnNewButton)
            )
);
gl_contentPane.setVerticalGroup(
        gl_contentPane.createParallelGroup(Alignment.CENTER)

.addGroup(gl_contentPane.createSequentialGroup()
            .addGap(19)

.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
            .addComponent(disease,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE,
GroupLayout.PREFERRED_SIZE)
            .addComponent(lblEnterName))
            .addGap(33)

.addGroup(gl_contentPane.createParallelGroup()
            .addComponent(btnDelete)

```

```

        .addComponent(btnNewButton)
        ));
        contentPane.setLayout(gl_contentPane);
    }
}

```

Manage diseases:

```

public class ManageDiseases extends JFrame {
    private static final long serialVersionUID = 1L;
    static ManageDiseases frame;
    private JPanel contentPane;

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

    /**
     * Create the frame.
     */
    public ManageDiseases() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 450, 433);
        contentPane = new JPanel();
        contentPane.setForeground(Color.GRAY);
    }
}

```

```
contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
setContentPane(contentPane);
```

```
JLabel lblSection = new JLabel("Manage Diseases");
lblSection.setFont(new Font("Tahoma", Font.PLAIN, 22));
```

```
JButton btnAdd = new JButton("Add Disease");
btnAdd.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        AddDisease.main(new String[]{});
        frame.dispose();
    }
});
btnAdd.setFont(new Font("Tahoma", Font.PLAIN, 13));
```

```
JButton btnSUpdate = new JButton("Modify Disease Deatils");
btnSUpdate.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent arg0) {
        ModifyDiseases.main(new String[]{});
        frame.dispose();
    }
});
btnSUpdate.setFont(new Font("Tahoma", Font.PLAIN, 13));
```

```
JButton btnSDelete = new JButton("Delete Disease");
btnSDelete.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        DeleteDisease.main(new String[]{});
        frame.dispose();
    }
});
btnSDelete.setFont(new Font("Tahoma", Font.PLAIN, 13));
```

```
JButton btnView = new JButton("View All Diseases");
btnView.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
```



```

        ViewDiseases.main(new String[]{});
    }
});
btnView.setFont(new Font("Tahoma", Font.PLAIN, 13));

JButton btnBack = new JButton("Back");
btnBack.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Menu.main(new String[]{});
        frame.dispose();
    }
});
btnBack.setFont(new Font("Tahoma", Font.PLAIN, 13));

GroupLayout gl_contentPane = new GroupLayout(contentPane);
gl_contentPane.setHorizontalGroup(

    gl_contentPane.createParallelGroup(Alignment.LEADING)
        .addGroup(Alignment.TRAILING,
gl_contentPane.createSequentialGroup()
            .addContainerGap(81, Short.MAX_VALUE)
            .addComponent(lblSection)
            .addGap(54))

        .addGroup(gl_contentPane.createSequentialGroup()
            .addGap(132)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
            .addComponent(btnBack,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
            .addComponent(btnView,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
            .addComponent(btnSDelete,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
            .addComponent(btnSUpdate,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
            .addComponent(btnAdd,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE))

```

```

        .addContainerGap(101,
Short.MAX_VALUE))
        );
        gl_contentPane.setVerticalGroup(

        gl_contentPane.createParallelGroup(Alignment.LEADING)

        .addGroup(gl_contentPane.createSequentialGroup()
                .addContainerGap()
                .addComponent(lblSection)
                .addGap(18)
                .addComponent(btnAdd,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                .addGap(18)
                .addComponent(btnSUpdate,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                .addGap(18)
                .addComponent(btnSDelete,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                .addGap(18)
                .addComponent(btnView,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                .addGap(18)
                .addComponent(btnBack,
 GroupLayout.PREFERRED_SIZE, 37, GroupLayout.PREFERRED_SIZE)
                .addContainerGap(16, Short.MAX_VALUE))
        );
        contentPane.setLayout(gl_contentPane);
    }

}

```

Modify Diseases:

```

public class ModifyDiseases extends JFrame {

    private static final long serialVersionUID = 1L;
    static ModifyDiseases frame;
    private JPanel contentPane;
    private JTextField diseaseNameLbl;

```

```

private JLabel diseaseName;
private JTextField discriptionLbl;
private JLabel disease;
private JTextField medicationLbl;
private JLabel medication;

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

/**
 * Create the frame.
 */
public ModifyDiseases() {
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(110, 95, 650, 350);
    setResizable(false);

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

```

```

diseaseNameLbl = new JTextField();
diseaseNameLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
diseaseNameLbl.setBounds(290, 37, 326, 40);
contentPane.add(diseaseNameLbl);
diseaseNameLbl.setColumns(10);

discriptionLbl = new JTextField();
discriptionLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
discriptionLbl.setBounds(290, 87, 326, 40);
contentPane.add(discriptionLbl);
diseaseNameLbl.setColumns(10);

medicationLbl = new JTextField();
medicationLbl.setFont(new Font("Tahoma", Font.PLAIN, 18));
medicationLbl.setBounds(290, 137, 326, 40);
contentPane.add(medicationLbl);
medicationLbl.setColumns(10);

diseaseNameLbl.setText("");
    discriptionLbl.setText("");
    medicationLbl.setText("");

    Connection con = DB.getConnection();

    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            frame.dispose();
            ManageSymptoms.main(new String[]{});
        }
    });
    btnBack.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnBack.setBackground(new Color(240, 240, 240));
    btnBack.setBounds(450, 225, 130, 50);
    contentPane.add(btnBack);
    JButton btnSave = new JButton("Save");
    btnSave.addActionListener(new ActionListener() {

```

```

public void actionPerformed(ActionEvent e) {
    String b = diseaseNameLbl.getText();
    String t = discriptionLbl.getText();
    String a = medicationLbl.getText();
    if(DiseaseDetails.checkDisease(b)) {
        try {
            PreparedStatement st = con.prepareStatement("Update disease
set descr:= ?, medicine= ? where name=?");
            st.setString(1, t);
            st.setString(2, a);
            st.setString(3, b);
            st.executeUpdate();
            System.out.println("updated info for : " + b);
            JOptionPane.showMessageDialog(btnSave, "Data Updated
Successfully :)");

        } catch (SQLException sqlException) {
            sqlException.printStackTrace();
        }
        else{

            JOptionPane.showMessageDialog(ModifyDiseases.this,"Disease
doesnot exist !!!");

            diseaseNameLbl.setText("");
            discriptionLbl.setText("");
            medicationLbl.setText("");

        }

    }

});
btnSave.setFont(new Font("Tahoma", Font.PLAIN, 20));
btnSave.setBackground(new Color(240, 240, 240));
btnSave.setBounds(300, 225, 130, 50);
contentPane.add(btnSave);

diseaseName = new JLabel("Disease name :");

```

```
diseaseName.setFont(new Font("Tahoma", Font.PLAIN, 18));
diseaseName.setBounds(45, 37, 326, 40);
contentPane.add(diseaseName);
```

```
disease = new JLabel("Description :");
disease.setFont(new Font("Tahoma", Font.PLAIN, 18));
disease.setBounds(45, 87, 326, 40);
contentPane.add(disease);
```

```
medication = new JLabel("Suggested Medicine :");
medication.setFont(new Font("Tahoma", Font.PLAIN, 18));
medication.setBounds(45, 137, 326, 40);
contentPane.add(medication);
```

```
JButton btnLoad = new JButton("Load");
btnLoad.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String book = diseaseNameLbl.getText();
        String title = discriptionLbl.getText();
        String author = medicationLbl.getText();
        Connection con = DB.getConnection();
        if(DiseaseDetails.checkDisease(book)) {
            Statement st;
            try {
                System.out.println("hi");
                PreparedStatement stmt = con.prepareStatement("select name,
descr, medicine from disease where disease.name=? ");
                stmt.setString(1,book);
                ResultSet rs = stmt.executeQuery();
                while(rs.next())
                {diseaseNameLbl.setText(rs.getString(1));
                 discriptionLbl.setText(rs.getString(2));
                 medicationLbl.setText(rs.getString(3));}
            } catch (SQLException e1) {
```

```

        e1.printStackTrace();
    }else {

        JOptionPane.showMessageDialog(ModifyDiseases.this,"Dsiease name
is Invalid !!!");

        diseaseNameLbl.setText("");
        discriptionLbl.setText("");
        medicationLbl.setText("");
    }

    });
    btnLoad.setFont(new Font("Tahoma", Font.PLAIN, 20));
    btnLoad.setBackground(new Color(240, 240, 240));
    btnLoad.setBounds(150, 225, 130, 50);
    contentPane.add(btnLoad);

}
}

```

View disease:

```

public class ViewDiseases extends JFrame {

    private JPanel contentPane;
    private.JTable table;
    static ViewDiseases frame = new ViewDiseases();

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    frame.setTitle("Diseases");
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

```

    }
    });
}

/**
 * Create the frame.
 */
public ViewDiseases() {
    setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    setBounds(100, 100, 450, 300);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    contentPane.setLayout(new BorderLayout(0, 0));
    setContentPane(contentPane);
    JButton btnBack = new JButton("Back");
    btnBack.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            Menu.main(new String[]{});
            frame.dispose();
        }
    });
    btnBack.setFont(new Font("Tahoma", Font.PLAIN, 13));

    String data[][]=null;
    String column[]=null;
    try{
        Connection con=DB.getConnection();
        PreparedStatement ps=con.prepareStatement("select * from
disease",ResultSet.TYPE_SCROLL_SENSITIVE,ResultSet.CONCUR_UPDA
TABLE);
        ResultSet rs=ps.executeQuery();

        ResultSetMetaData rsmd=rs.getMetaData();
        int cols=rsmd.getColumnCount();
        column=new String[cols];

```



```

        for(int i=1;i<=cols;i++){
            column[i-1]=rsmd.getColumnName(i);
        }

        rs.last();
        int rows=rs.getRow();
        rs.beforeFirst();

        data=new String[rows][cols];
        int count=0;
        while(rs.next()){
            for(int i=1;i<=cols;i++){
                data[count][i-1]=rs.getString(i);
            }
            count++;
        }
        con.close();
    }catch(Exception e){System.out.println(e);}

    table = new JTable(data,column);
    table.setDefaultEditor(Object.class, null);
    JScrollPane sp=new JScrollPane(table);

    contentPane.add(sp, BorderLayout.CENTER);
}

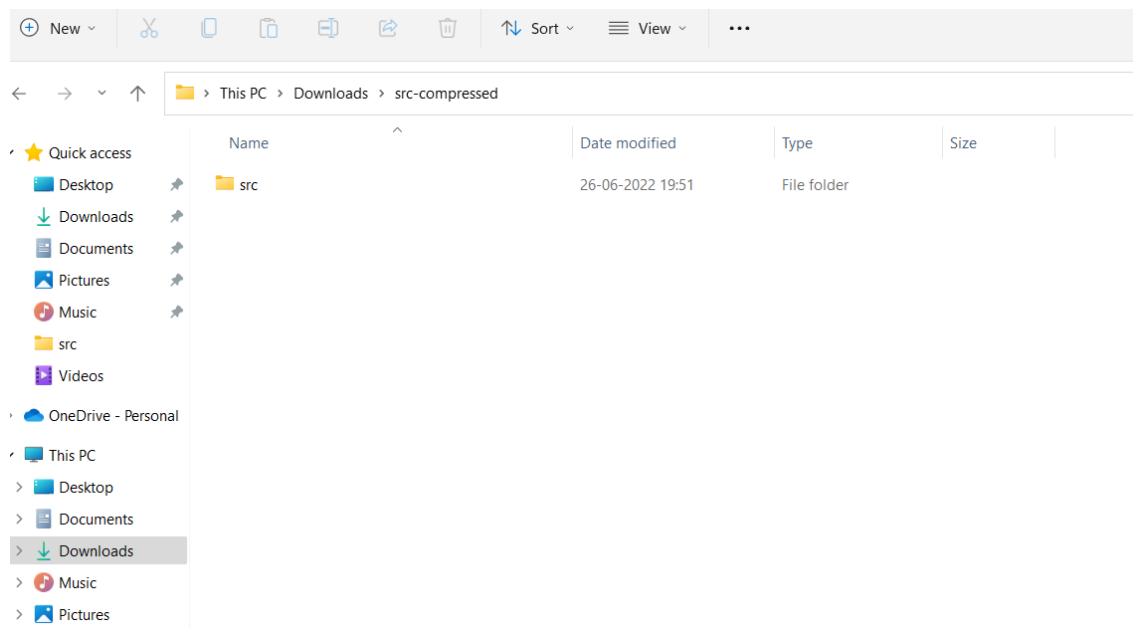
}

```

## Github links and folder structure:

Link: [Vamshidhar07/cognitive-disorders \(github.com\)](https://github.com/Vamshidhar07/cognitive-disorders)

## Folder structure:




src					
<div> <div> <div>New</div> <div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> </div> </div> <div>Sort</div> <div>View</div> <div></div> </div>					
<div> <div> <div></div> <div></div> <div></div> <div></div> </div> <div> <div>This PC</div> <div>Downloads</div> <div>src-compressed</div> <div>src</div> </div> </div>					
<div> <div>Quick access</div> <div> <div>Desktop</div> <div>Downloads</div> <div>Documents</div> <div>Pictures</div> <div>Music</div> <div>src</div> <div>Videos</div> </div> <div> <div>OneDrive - Personal</div> <div>This PC</div> <div>Desktop</div> <div>Documents</div> <div>Downloads</div> <div>Music</div> <div>Pictures</div> <div>Videos</div> <div>Windows (C:)</div> <div>New Volume (D:)</div> <div>New Volume (E:)</div> <div>Network</div> </div> </div>	Name	Date modified	Type	Size	
	AddChild	26-06-2022 19:51	IntelliJ IDEA	6 KB	
	AddDisease	26-06-2022 19:51	IntelliJ IDEA	6 KB	
	AddSymptom	26-06-2022 19:51	IntelliJ IDEA	6 KB	
	ChildDetails	26-06-2022 19:51	IntelliJ IDEA	2 KB	
	DB	26-06-2022 19:51	IntelliJ IDEA	1 KB	
	DeleteChild	26-06-2022 19:51	IntelliJ IDEA	4 KB	
	DeleteDisease	26-06-2022 19:51	IntelliJ IDEA	4 KB	
	DeleteSymptom	26-06-2022 19:51	IntelliJ IDEA	4 KB	
	DiseaseDetails	26-06-2022 19:51	IntelliJ IDEA	2 KB	
	ManageChild	26-06-2022 19:51	IntelliJ IDEA	5 KB	
	ManageDiseases	26-06-2022 19:51	IntelliJ IDEA	5 KB	
	ManageSymptoms	26-06-2022 19:51	IntelliJ IDEA	5 KB	
	Menu	26-06-2022 19:51	IntelliJ IDEA	4 KB	
	ModifyChild	26-06-2022 19:51	IntelliJ IDEA	6 KB	
	ModifyDiseases	26-06-2022 19:51	IntelliJ IDEA	7 KB	
	ModifySymptoms	26-06-2022 19:51	IntelliJ IDEA	8 KB	
	SymptomDetails	26-06-2022 19:51	IntelliJ IDEA	2 KB	
	ViewChild	26-06-2022 19:51	IntelliJ IDEA	3 KB	
	ViewDiseases	26-06-2022 19:51	IntelliJ IDEA	3 KB	

## TESTING:

Output screenshots:





 Book Info. — □ ×

Add Child

Name :

Age :

Symptom :

Submit

Back

Book Info.


### Add Child

Name :

Age :

Symptom :

Message

 Child added successfully!

Modify Child Info.

Child name :

Age :

Symptom :

Modify Child Info.

Child name : Vamshi

Age : 10

Symptom :

Message

Data Updated Successfully :)

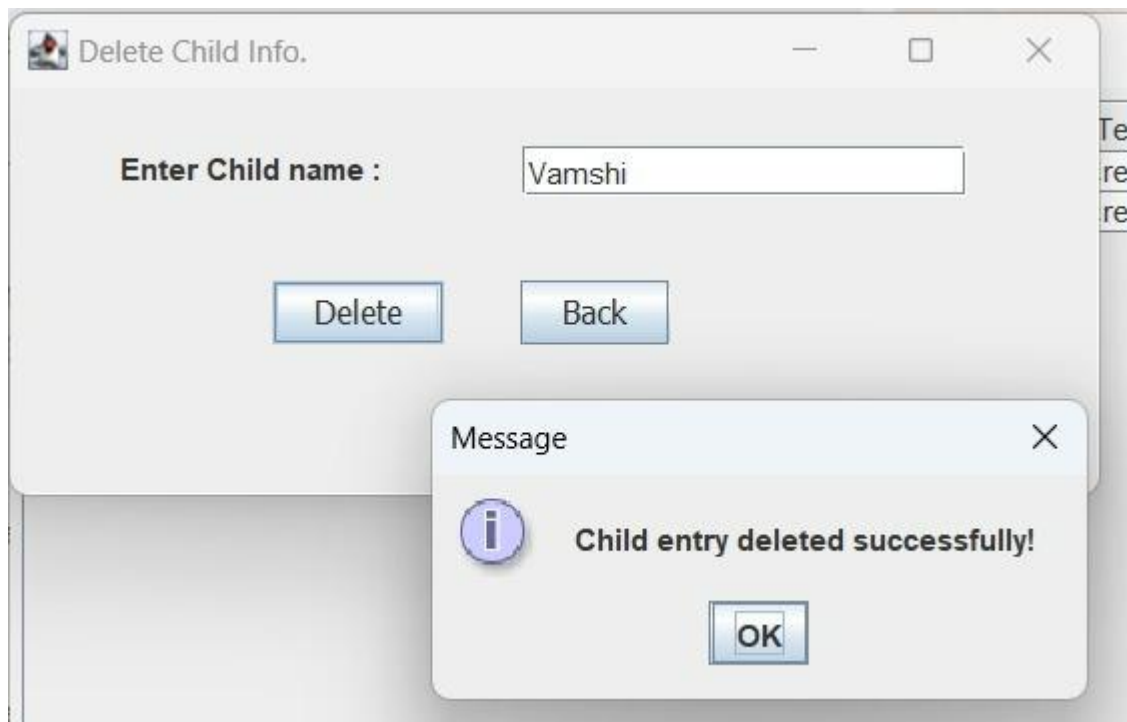
OK

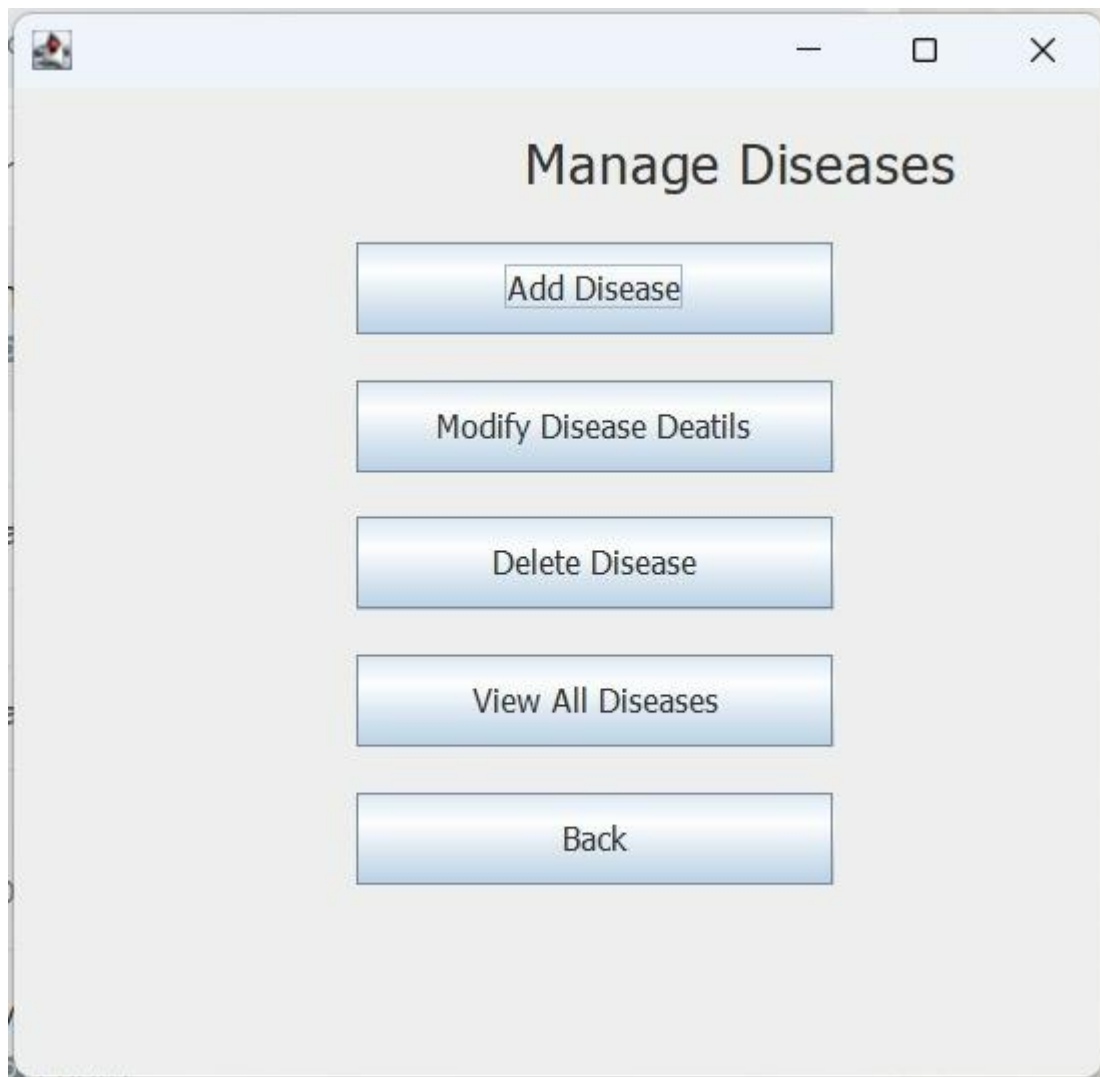
Load Save Back


Children

Name	age	symptom	Test	Disease	Description	Medicine
Chandu	11	temperature	thermal screening	Cough	desc	Dolo
Vamshi	10	temperature	thermal screening	Cough	desc	Dolo







 Disease Info. — □ ×

### Add Disease


**Disease Name :**

**Description :**


**Prescribed medicine :**

 Delete Symptom Info. — □ ×

**Enter Symptom name :**

 Diseases

disease_id	name	descr	medicine
2	Cough	desc	Dolo

 Book Info.

Add Symptom

Symptom Name :

Description :

Disease :

Submit

Back

Modify Symptoms Info.

Symptom name :

Suspected Disease :

Suggested Test :

Load Save Back

Delete Symptom Info.

Enter Symptom name :

Delete Back

## **CONCLUSION:**

Through this project we were able to achieve the ease of maintaining children database were suffering with various cognitive diseases that are listed in our database. We were able to track add and make changes to the data that we have stored. They were able to load the data related to a child along with the symptoms and the disease he/she is suffering with and also the medication required for the disease.

## **REFERENCES:**

1. Abraham Silberschatz, Henry F Korth, S. Sudarshan, Database System Concepts, 6th Edition, McGraw-Hill International Edition, 2010.
2. [https://docs.oracle.com/cd/E11882\\_01/server.112/e40540/intro.htm#CNCPT001](https://docs.oracle.com/cd/E11882_01/server.112/e40540/intro.htm#CNCPT001)
3. <https://docs.oracle.com/javase/8/docs/>