JAVA SWING BASED – COVID VACCINATION INFORMATION TRACKER DATABASE-SQL CONNECTIVITY USING JDBC

A

Report

Submitted in partial fulfilment of the
Requirements for the award of the Degree of
BACHELOR OF TECHNOLOGY
IN

INFORMATION TECHNOLOGY

By

A.Sreenija <1602-20-737-044>

Under the Guidance of

B. Leelavathy

Department of Information Technology

Vasavi College of Engineering (Autonomous)

(Affiliated to Osmania University)

Ibrahimbagh, Hyderabad-31

2021-2022

BONAFIDE CERTIFICATE

This to Certify that the project report titled **"COVID VACCINATION INFORMATION TRACKER DATABASE" project work of Ms.A.SREENIJA bearing Roll.no:1602-20-737-044 who carried out this project under my supervision in the IV semester for the academic year 2021-2022.**

Signature
external examiner

Signature
internal examiner

ABSTRACT:

This is project “Covid Vaccination Information Tracker Database “. we have to track the Number of slots available, type of vaccine taken and test cases that ensures the user has already taken dose1 or dose2 or booster. This is helpful as it shows the information about number of slots so that the user can arrive when his turn comes and it helps the user to find which type of vaccine he has taken and also the details about number of doses taken. To implement this and to display all the data on the screen we need to build the database with details of number of slots available as backend database. To build this portal, we are using SQL for the backend and java for the frontend part.

Introduction:

REQUIREMENT ANALYSIS

List of tables:

- User_details
- Slots
- Vaccine_type
- Doses_taken

List of attributes with their domain types:

User_details:

- sname(varchar2)
- email(varchar2)
- phone_no(number)
- aadhar_no(number)
- age(number)

Slots

- time(varchar2)
- duration(sysdate)
- aadhar_no(number)
- area(varchar2)

Vaccine_type

- sname(varchar2)
- aadhar_no(number)
- Type(varchar2)

Doses_taken

- first_dose
- second_dose
- booster_dose
- aadhar_no

THROUGH THE PROJECT:

This project helps to store data in a efficient way and it can be achieved through various sql commands and we can also store this for any future use and also we can save our data in a many different areas so we cannot lost all the data at once. The Covid Vaccination details are must as to know the number of doses and type of vaccine taken by a person. These project stores details of user, vaccine type and doses taken in database so that whenever it is necessary to know it would be easy for us to access the data.

ARCHITECTURE AND TECHNOLOGY USED:

SOFTWARE USED:

Java Eclipse, Oracle 11g Database, Java SE version 8, MYSQL .

Java SWING:

Swing is a GUI widget toolkit for Java. It is part of Oracle's Java Foundation Classes (JFC) – an API for providing a graphical user interface (GUI) for Java programs.

Swing was developed to provide a more sophisticated set of GUI components than the earlier AWT. Swing provides a look and feel that emulates the look and feel of several platforms, and also supports a pluggable look and feel that allows applications to have a look and feel unrelated to the underlying platform. It has more powerful and flexible components than AWT. In addition to familiar components such as buttons, check boxes and labels, Swing provides several advanced components such as tabbed panel, scroll panes, trees, tables, and lists.

SQL:

Structure Query Language (SQL) is a database query language used for storing and managing data in Relational DBMS. SQL was the first commercial language introduced for E.F Codd's **Relational** model of database. Today almost all RDBMS (MySql, Oracle, Infomix, Sybase, MS Access) use **SQL** as the standard database query language. SQL is used to perform all types of data operations in RDBMS.

Java-SQL Connectivity using JDBC:

Java Database Connectivity (**JDBC**) is an application programming interface (API) for the programming language Java, which defines how a client may access a database. It is a Java-based data access technology used for Java database connectivity. It is part of the Java Standard Edition platform, from Oracle Corporation. It provides methods to query and update data in a database and is oriented towards relational databases.

The connection to the database can be performed using Java programming (JDBC API) as:

```
  try
  {
        Class.forName("oracle.jdbc.driver.OracleDriver");
 }

 catch (Exception e)
{
        System.err.println("Unable to find and load driver");
        System.exit(1);
}
public void connectToDB()
  {
                try
                {
                 connection =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1522:ORCL","mydbms","mydbms");
                 statement = connection.createStatement();

                }
                catch (SQLException connectException)
                {
                 System.out.println(connectException.getMessage());
                 System.out.println(connectException.getSQLState());
                 System.out.println(connectException.getErrorCode());
                 System.exit(1);
                }
  }
```

Thus, the connection from Java to Oracle database is performed and therefore, can be used for updating tables in the database directly.

Table Created in SQL for above mentioned purpose is as:

1. create table user_details(
   sname varchar2(30),
   phone_no number,
   address varchar2(40),
   aadhar_no number,
   age number);

Setting aadhar_no as primary key:
Alter table user_details add primary key(aadhar_no);

Select C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19_8\i

```
SQL> desc user_details;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 SNAME                                              VARCHAR2(20)
 PHONENO                                            NUMBER(10)
 ADDRESS                                            VARCHAR2(40)
 AADHARNO                                  NOT NULL NUMBER(20)
 AGE                                                NUMBER(5)
```

2. create table slots(
   area varchar2(20),
   center_code number,
   duration date,
   time number);

   Setting aadhar_no as foreign key.

C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19_8\instantclient_1

```
SQL> create table slots(
  2  area varchar2(20),
  3  center_code number(10),
  4  duration date,
  5  time number(10),
  6  aadharno number(20));

Table created.

SQL> alter table slots add foreign key(aadharno) references user_details;

Table altered.
```

C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19_8\instantclient_19_8\sqlplus.exe

```
SQL> desc slots;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AREA                                               VARCHAR2(20)
 CENTER_CODE                               NOT NULL NUMBER(10)
 DURATION                                           DATE
 TIME                                               NUMBER(10)
 AADHARNO                                           NUMBER(20)

SQL>
```

3. create table vaccine_type(
   sname varchar2(30)),
   type varchar2(20),
   aadhar_no number references user_details(aadhar_no));

C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19

```
SQL> create table vaccine_type(
  2  sname varchar2(30),
  3  type varchar2(20),
  4  aadharno number(20) references user_details(aadharno));

Table created.
```

C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19_8\instantclient_19_8\sqlplu

```
SQL> desc vaccine_type;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 SNAME                                              VARCHAR2(30)
 TYPE                                               VARCHAR2(20)
 AADHARNO                                           NUMBER(20)

SQL>
```

4. create table doses_taken(
   aadhar_no number references user_details(aadhar_no),
   first_dose number,
   second_dose number,
   booster_dose number);

C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient

```
SQL> create table doses_taken(
  2  aadharno number(20) references user_details(aadharno),
  3  first_dose number(10),
  4  second_dose number(10),
  5  booster_dose number(10));

Table created.
```

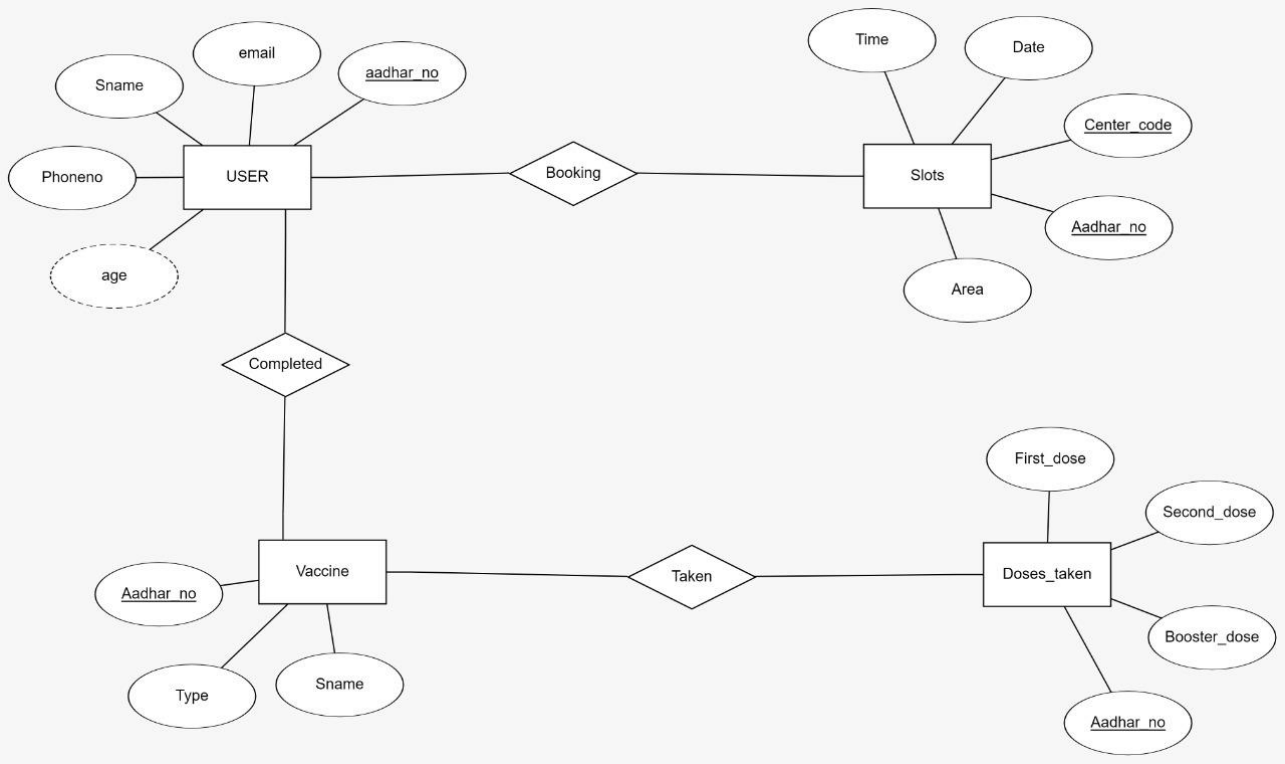
C:\Users\91891\Downloads\1602-20-737-044\instaclient\instaclient\instantclient_19_8\instantclient_19_8\sqlplus

```
SQL> desc doses_taken;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 AADHARNO                                           NUMBER(20)
 FIRST_DOSE                                         NUMBER(10)
 SECOND_DOSE                                        NUMBER(10)
 BOOSTER_DOSE                                       NUMBER(10)

SQL>
```

DESIGN:

ER DIAGRAM:



Implementation:

Program:

Menu:

```
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("Covid Vaccination Tracker Database");
					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("Covid Tracker ");
		lblLibrarianSection.setFont(new Font("Academy Engraved LET", Font.BOLD , 40));

		JButton btnBInfo = new JButton("User Details");
		btnBInfo.addActionListener(new ActionListener() {
```

```
                        public void actionPerformed(ActionEvent e) {
                        AddUser.main(new String[]{});
                        frame.dispose();
                        }
                });
                btnBInfo.setFont(new Font("Tahoma", Font.PLAIN, 13));

                JButton btnSInfo = new JButton("Slots");
                btnSInfo.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent arg0) {
                                SLOTSUI.main(new String[]{});
                                frame.dispose();
                        }
                });
                btnSInfo.setFont(new Font("Tahoma", Font.PLAIN, 13));

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

                JButton btnReturnBook = new JButton("Doses Taken");
                btnReturnBook.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                                DosesUI.main(new String[]{});
                                frame.dispose();
                        }
                });
                btnReturnBook.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(lblLibrarianSection)
                                        .addGap(54))
                                .addGroup(gl_contentPane.createSequentialGroup()
                                        .addGap(132)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                .addComponent(btnReturnBook,
GroupLayout.PREFERRED_SIZE, 191, GroupLayout.PREFERRED_SIZE)
                                                .addComponent(btnIssueBook,
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(btnIssueBook, GroupLayout.PREFERRED_SIZE,
37, GroupLayout.PREFERRED_SIZE)
                                .addGap(18)
                                .addComponent(btnReturnBook, GroupLayout.PREFERRED_SIZE,
37, GroupLayout.PREFERRED_SIZE)
                                .addGap(18)
                                .addGap(18)
                                .addContainerGap(16, Short.MAX_VALUE))
                );
                contentPane.setLayout(gl_contentPane);
        }

}
```

AddUser:

```
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 javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;


public class AddUser extends JFrame {
        private static final long serialVersionUID = 1L;
        static AddUser frame;
        private JPanel contentPane;
        private JTextField textField;
        private JTextField textField_1;
        private JTextField textField_2;
        private JTextField textField_3;
        private JTextField textField_4;

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

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

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

                JLabel lblEnterDetails = new JLabel("Enter Details");
                lblEnterDetails.setForeground(Color.GRAY);
                lblEnterDetails.setFont(new Font("Tahoma", Font.PLAIN, 18));

                JLabel lblusername = new JLabel("User name :");

                JLabel lblphone = new JLabel("Phone no. :");

                JLabel lblaadhar = new JLabel("Aadhar no. :");

                JLabel lbladdress = new JLabel("Address :");

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

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

                textField_1 = new JTextField();
                textField_1.setColumns(10);

                textField_2 = new JTextField();
                textField_2.setColumns(10);

                textField_3 = new JTextField();
                textField_3.setColumns(10);

                textField_4 = new JTextField();
                textField_4.setColumns(10);

                JButton btnAddUser = new JButton("Insert");
                btnAddUser.addActionListener(new ActionListener() {
                        public void actionPerformed(ActionEvent e) {
                        String sname=textField.getText();
                        String phone_no=textField_1.getText();
                        String aadhar_no=textField_2.getText();
                        String address=textField_3.getText();
                        String age=textField_4.getText();
                        long phone=Long.parseLong(phone_no);
                        long aadhar=Long.parseLong(aadhar_no);
                        int ages=Integer.parseInt(age);
                        int i=Userdetails.insertbook(sname, phone, aadhar,address,ages);
                        if(i>0){
                                JOptionPane.showMessageDialog(AddUser.this,"User added
successfully!");
```

```
                    }else{
                          JOptionPane.showMessageDialog(AddUser.this,"Unknown Error
!!!\nInsertion not completed");
                    }
                    }

             });

             JButton btnBack = new JButton("Delete");
             btnBack.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e1) {
                          String sname=textField_2.getText();
                          long L1= Long.parseLong(sname);
                          Userdetails.deletebook(L1);
                          frame.dispose();
                    }
                    });

             JButton btnNext = new JButton("Next");
             btnNext.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e1) {
                          String aadhar_no = textField_2.getText();
                          Dosestaken.DosesTaken(aadhar_no);
                          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(lblEnterDetails))
                                        .addGroup(gl_contentPane.createSequentialGroup()
                                              .addGap(18)

       .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                    .addComponent(lblusername)
                                                    .addComponent(lblphone)
                                                    .addComponent(lblaadhar)
                                                    .addComponent(lbladdress)
                                                    .addComponent(lblage)
                                                    )
                                              .addGap(47)

       .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)


                                                    .addComponent(textField_4,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
                                                    .addComponent(textField_3,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
                                                    .addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
                                                    .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
                                                    .addComponent(textField,
```

```
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))
                                .addContainerGap(125, Short.MAX_VALUE))
                            .addGroup(Alignment.LEADING,
gl_contentPane.createSequentialGroup()
                                .addGap(50)
                                .addComponent(btnAddUser)
                                .addGap(30)
                                .addComponent(btnBack)
                                .addGap(30)
                                .addComponent(btnNext)
                                .addContainerGap())
            );
            gl_contentPane.setVerticalGroup(
                    gl_contentPane.createParallelGroup(Alignment.LEADING)
                            .addGroup(gl_contentPane.createSequentialGroup()
                                .addComponent(lblEnterDetails)
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblusername)
                                        .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblphone)
                                        .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblaadhar)
                                        .addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                            .addComponent(lbladdress)
                                            .addComponent(textField_3,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                        .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblage)
                                                .addComponent(textField_4,
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(btnAddUser,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

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

```
                                                .addComponent(btnNext,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE))

	);
		contentPane.setLayout(gl_contentPane);
	}

}
```

Userdetails:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Userdetails {

	public static boolean checkbook(String sname){
		boolean status=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from
user_details where sname=?");
			ps.setString(1,sname);
		    ResultSet rs=ps.executeQuery();
			status=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return status;
	}
	public static int insertbook(String sname,long phone_no,long aadhar_no,String
address, int age){
		int status=0;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("insert into
user_details(sname,phone_no,aadhar_no,address,age) values(?,?,?,?,?)");
			ps.setString(1,sname);
			String s1= Long.toString(phone_no);
			ps.setString(2,s1);
			String s2= Long.toString(aadhar_no);
			ps.setString(3,s2);
			ps.setString(4, address);
			String s3= Integer.toString(age);
			ps.setString(5,s3);
			status=ps.executeUpdate();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return status;
	}
	public static int deletebook(long id){
		int status=0;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("delete from user_details
where aadhar_no=?");
			String s1=Long.toString(id);
			ps.setString(1,s1);
			status=ps.executeUpdate();
			con.close();
```

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

Checkdoses:

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.Statement;
//import com.mysql.cj.xdevapi.Statement;
public class checkdoses {
	public static boolean checklogin(String id){
		boolean allow=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from
doses_taken where aadhar_no =?");
			ps.setString(1,id);
		    ResultSet rs=ps.executeQuery();
			allow=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return allow;
	}
	/*public static boolean checkpassword(String id,String pass){
		boolean state=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from Slots
where center_code =? and Area=?");
			ps.setString(1,id);
			ps.setString(2,pass);
		    ResultSet rs=ps.executeQuery();
			state=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return state;
	}*/
	public static int NumSlots(String id){
		//boolean state=false;
		int sum = 0;
		try{
			//Connection con=DB.getConnection();
			 Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/covid","root","srinija");
			//PreparedStatement ps=con.prepareStatement("select
first_dose+second_dose+booster_dose from doses_taken where aadhar_no = ?");
			Statement s1= con.createStatement();
			//ps.setString(1,id);
			//ps.setString(2,pass);
			//String str = rs
		    ResultSet s=s1.executeQuery("select
first_dose,booster_dose,second_dose from doses_taken where aadhar_no = ?");
		    //String str = rs.getString("time");
		    //num = Integer.parseInt("time");
```

```
                //int sum=0;
                //while(s.next()) {

      sum=s.getInt("first_dose")+s.getInt("second_dose")+s.getInt("booster_dose");
                //}
                   //state=rs.next();
                   con.close();
             }catch(Exception e){System.out.println(e);}
             return sum;
      }

}
```

Checkvaccine:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class checkvaccine {
      public static boolean checklogin(String id){
             boolean allow=false;
             try{
                    Connection con=DB.getConnection();
                    PreparedStatement ps=con.prepareStatement("select * from
vaccine_type where aadhar_no =?");
                    ps.setString(1,id);
                 ResultSet rs=ps.executeQuery();
                    allow=rs.next();
                    con.close();
             }catch(Exception e){System.out.println(e);}
             return allow;
      }
      public static boolean NumSlots(String id){
             boolean state=false;
             try{
                    Connection con=DB.getConnection();
                    PreparedStatement ps=con.prepareStatement("select * from
vaccine_type where aadhar_no = ?");
                    ps.setString(1,id);
                 ResultSet rs=ps.executeQuery();
                    state=rs.next();
                    con.close();
             }catch(Exception e){System.out.println(e);}
             return state;
      }

}
```

Checkvaccinetype:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class checkvaccinetype {
      public static boolean checklogin(String id){
             boolean allow=false;
```

```
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from Slots
where center_code =?");
			ps.setString(1,id);
		    ResultSet rs=ps.executeQuery();
			allow=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return allow;
	}
	public static boolean checkpassword(String id,String pass){
		boolean state=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from Slots
where center_code =? and Area=?");
			ps.setString(1,id);
			ps.setString(2,pass);
		    ResultSet rs=ps.executeQuery();
			state=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return state;
	}
	public static boolean NumSlots(String id,String pass){
		boolean state=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from Slots
where center_code = ?,and Area=?");
			ps.setString(1,id);
			ps.setString(2,pass);
		    ResultSet rs=ps.executeQuery();
			state=rs.next();
			con.close();
		}catch(Exception e){System.out.println(e);}
		return state;
	}


}
```

Dosesdetails:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Dosesdetails {

	public static boolean checkbook(String bookid){
		boolean status=false;
		try{
			Connection con=DB.getConnection();
			PreparedStatement ps=con.prepareStatement("select * from
doses_taken where aadhar_no=?");

			ps.setString(1,bookid);
		    ResultSet rs=ps.executeQuery();
```

```
                        status=rs.next();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return status;
        }
        public static int insertbook(String aadhar_no,String first_dose,String
second_dose,String booster_dose, String type){
                int status=0;
                int status1=0;
                try{
                        Connection con=DB.getConnection();
                        PreparedStatement ps=con.prepareStatement("insert into
doses_taken(aadhar_no,first_dose,second_dose,booster_dose) values(?,?,?,?)");
                        PreparedStatement ps1=con.prepareStatement("insert into
vaccine_type(aadhar_no,type) values(?,?)");
                        //String s1 = Integer.toString(first_dose);
                        //String s2 = Integer.toString(second_dose);
                        //String s3 = Integer.toString(booster_dose);
                        ps.setString(1,aadhar_no);
                        ps.setString(2,first_dose);
                        ps.setString(3,second_dose);
                        ps.setString(4,booster_dose);
                        ps1.setString(1,aadhar_no);
                        ps1.setString(2,type);
                        status=ps.executeUpdate();
                        status1 = ps1.executeUpdate();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return status & status1;
        }
        public static int deletebook(String id){
                int status=0;
                try{
                        Connection con=DB.getConnection();
                        PreparedStatement ps=con.prepareStatement("delete from book where
bookid=?");
                        ps.setString(1,id);
                        status=ps.executeUpdate();
                        con.close();
                }catch(Exception e){System.out.println(e);}
                return status;
        }
}
```

Dosestaken:

```
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 javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
```

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
//import javax.swing.LayoutStyle.ComponentPlacement;

public class Dosestaken extends JFrame {
	private static final long serialVersionUID = 1L;
	static Dosestaken frame;
	private JPanel contentPane;
	private JTextField textField;
	private JTextField textField_1;
	private JTextField textField_2;
	private JTextField textField_3;

	/**
	 * Launch the application.
	 */
	public static void DosesTaken(String aadhar_no) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Dosestaken(aadhar_no);
					frame.setTitle("Vaccine Details");
					frame.setVisible(true);
				} catch (Exception e) {
					e.printStackTrace();
				}
			}
		});
	}

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

		JLabel lblDosetaken = new JLabel("Vaccine Details");
		lblDosetaken.setForeground(Color.GRAY);
		lblDosetaken.setFont(new Font("Tahoma", Font.PLAIN, 18));

		JLabel lbltype = new JLabel("vaccine type :");

		JLabel lblfirst_dose = new JLabel("First Dose :");

		JLabel lblsecond_dose = new JLabel("Second Dose :");

		JLabel lblbooster_dose = new JLabel("Booster Dose :");

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

		textField_1 = new JTextField();
		textField_1.setColumns(10);

		textField_2 = new JTextField();
		textField_2.setColumns(10);
```

```
		textField_3 = new JTextField();
		textField_3.setColumns(10);

		JButton btnDosetaken = new JButton("Insert");
		btnDosetaken.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
			String type=textField.getText();
			String firstdose=textField_1.getText();
			String seconddose=textField_2.getText();
			String boosterdose=textField_3.getText();
			int i=Dosesdetails.insertbook(aadhar_no,firstdose, seconddose,
boosterdose, type);
			if(i>0){
				JOptionPane.showMessageDialog(Dosestaken.this,"Details added
successfully!");
			}else{
				JOptionPane.showMessageDialog(Dosestaken.this,"Unknown Error
!!!\nInsertion not completed");
			}
			}
			}
		);

		JButton btnBack = new JButton("Back");
		btnBack.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e1) {
				AddUser.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(lblDosetaken))
					.addGroup(gl_contentPane.createSequentialGroup()
						.addGap(18)

	.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
							.addComponent(lbltype)
							.addComponent(lblfirst_dose)
							.addComponent(lblsecond_dose)
							.addComponent(lblbooster_dose)
							)
					.addGap(47)

	.addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)

							.addComponent(textField_3,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
							.addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
							.addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE)
							.addComponent(textField,
```

```
GroupLayout.PREFERRED_SIZE, 167, GroupLayout.PREFERRED_SIZE))))
                                .addContainerGap(125, Short.MAX_VALUE))
                            .addGroup(Alignment.LEADING,
gl_contentPane.createSequentialGroup()
                                .addGap(50)
                                .addComponent(btnDosetaken)
                                .addGap(30)
                                .addComponent(btnBack)
                                .addContainerGap())
            );
            gl_contentPane.setVerticalGroup(
                  gl_contentPane.createParallelGroup(Alignment.LEADING)
                        .addGroup(gl_contentPane.createSequentialGroup()
                                .addComponent(lblDosetaken)
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                    .addComponent(lbltype)
                                    .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                    .addComponent(lblfirst_dose)
                                    .addComponent(textField_1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                    .addComponent(lblsecond_dose)
                                    .addComponent(textField_2,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                .addGap(18)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                    .addComponent(lblbooster_dose)
                                    .addComponent(textField_3,
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(btnDosetaken,
GroupLayout.PREFERRED_SIZE, 31, GroupLayout.PREFERRED_SIZE)

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

      )     );
            contentPane.setLayout(gl_contentPane);
      }

}
```

DosseUI:

```
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 javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
//import javax.swing.JPasswordField;

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

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

	/**
	 * Create the frame.
	 */
	public DosesUI() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(103, 95, 700, 750);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		JLabel lblid = new JLabel("Aadhar no :");

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

		JButton btnClose = new JButton("Close");
	  btnClose.addActionListener(new ActionListener() {
	      public void actionPerformed(ActionEvent e) {
	        frame.dispose();
	      }});
```

```
        btnClose.setFont(new Font("Tahoma", Font.PLAIN, 12));
        btnClose.setBackground(new Color(240, 240, 240));
        btnClose.setBounds(500, 666, 80, 30);
        contentPane.add(btnClose);

             JButton btnLogin = new JButton("Submit");
             btnLogin.addActionListener(new ActionListener() {
                    public void actionPerformed(ActionEvent e) {
                    String name=textField.getText();
                    if(checkdoses.checklogin(name)){
                                 JOptionPane.showMessageDialog(DosesUI.this, " No.of
Doses: "+ checkdoses.NumSlots(name));
                           frame.dispose();
                    }
                           else{
                                 JOptionPane.showMessageDialog(DosesUI.this," Error!!!
\n Aadhar number is invalid");
                           textField.setText("");
                           }
                    }}
             );

             textField = new JTextField();
             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(15)
                                                      .addComponent(lblid)
                                               .addGap(30)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                      .addComponent(textField,
GroupLayout.DEFAULT_SIZE, 172, Short.MAX_VALUE))
                                               .addGap(10)

      .addGroup(gl_contentPane.createParallelGroup()
                                                             .addComponent(btnLogin))))
                                  )
             );
             gl_contentPane.setVerticalGroup(
                    gl_contentPane.createParallelGroup(Alignment.LEADING)
                      .addGroup(gl_contentPane.createSequentialGroup()
                                  .addGap(30)

      .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                        .addComponent(lblid)
                                        .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                  .addGap(20)
                                  .addGroup(gl_contentPane.createParallelGroup()
                                               .addComponent(btnLogin))
             ));
             contentPane.setLayout(gl_contentPane);
      }
}
```

SlotsUI:

```
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.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
//import javax.swing.JPasswordField;

public class SLOTSUI extends JFrame {
	private static final long serialVersionUID = 1L;
	static SLOTSUI frame;
	private JPanel contentPane;
	private JTextField textField;
	private JTextField textField1;

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

	/**
	 * Create the frame.
	 */
	public SLOTSUI() {
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setBounds(103, 95, 700, 750);
		contentPane = new JPanel();
		contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
		setContentPane(contentPane);

		JLabel lblid = new JLabel("Center_code :");

		JLabel lblpassword = new JLabel("Area :");

		textField = new JTextField();
		textField.setColumns(10);
```

```
             JButton btnClose = new JButton("Close");
          btnClose.addActionListener(new ActionListener() {
              public void actionPerformed(ActionEvent e) {
                frame.dispose();
              }});
          btnClose.setFont(new Font("Tahoma", Font.PLAIN, 12));
          btnClose.setBackground(new Color(240, 240, 240));
          btnClose.setBounds(500, 666, 80, 30);
          contentPane.add(btnClose);

               JButton btnLogin = new JButton("Submit");
               btnLogin.addActionListener(new ActionListener() {
                      public void actionPerformed(ActionEvent e) {
                      String name=textField.getText();
                      String password=textField1.getText();
                      if(checkvaccinetype.checklogin(name)){
                             if(checkvaccinetype.checkpassword(name,password)){
                                    JOptionPane.showMessageDialog(SLOTSUI.this, " Slots
Present");
                             frame.dispose();
                      }
                             else{
                                    JOptionPane.showMessageDialog(SLOTSUI.this, " Error!!!
\n Area doesn't exist");
                                    textField1.setText("");
                                    }
                             }
                             else{
                                    JOptionPane.showMessageDialog(SLOTSUI.this," Error!!!
\n Center code is invalid");
                             textField.setText("");
                             textField1.setText("");
                      }
                      }}
               );

               textField1 = new JTextField();
               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(19)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING)
                                                        .addComponent(lblid)
                                                        .addComponent(lblpassword))
                                                 .addGap(47)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                        .addComponent(textField1)
                                                        .addComponent(textField,
GroupLayout.DEFAULT_SIZE, 172, Short.MAX_VALUE))

        .addGroup(gl_contentPane.createParallelGroup()
                                                                .addComponent(btnLogin))))
                                          )
```

```
		);
		gl_contentPane.setVerticalGroup(
			gl_contentPane.createParallelGroup(Alignment.LEADING)
			  .addGroup(gl_contentPane.createSequentialGroup()
					.addGap(26)

	.addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
						.addComponent(lblid)
						.addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
					.addGap(28)

	.addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
						.addComponent(lblpassword)
						.addComponent(textField1,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
					.addGap(18)
					.addGroup(gl_contentPane.createParallelGroup()
							.addComponent(btnLogin))
		));
		contentPane.setLayout(gl_contentPane);
	}
}
```

Vaccine:

```
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.ImageIcon;
import javax.swing.JLabel;
import javax.swing.JOptionPane;

import java.awt.Font;
import java.awt.Color;
import javax.swing.JTextField;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
//import javax.swing.JPasswordField;

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

	/**
	 * Launch the application.
	 */
	public static void main(String[] args) {
		EventQueue.invokeLater(new Runnable() {
			public void run() {
				try {
					frame = new Vaccine();
```

```
                                frame.setTitle("Covid Vaccination Tracker Database");
                                frame.setVisible(true);
                        } catch (Exception e) {
                                e.printStackTrace();
                        }
                }
        });
    }

    /**
     * Create the frame.
     */
    public Vaccine() {
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(103, 95, 700, 750);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);

        JLabel lblid = new JLabel("Aadhar no :");

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

        JButton btnClose = new JButton("Close");
      btnClose.addActionListener(new ActionListener() {
          public void actionPerformed(ActionEvent e) {
            frame.dispose();
          }});
      btnClose.setFont(new Font("Tahoma", Font.PLAIN, 12));
      btnClose.setBackground(new Color(240, 240, 240));
      btnClose.setBounds(500, 666, 80, 30);
      contentPane.add(btnClose);

        JButton btnLogin = new JButton("Submit");
        btnLogin.addActionListener(new ActionListener() {
                public void actionPerformed(ActionEvent e) {
                String name=textField.getText();
                if(checkvaccine.checklogin(name)){
                                JOptionPane.showMessageDialog(Vaccine.this, " Vaccine
Type: "+ checkvaccine.NumSlots(name));
                        frame.dispose();
                }
                        else{
                                JOptionPane.showMessageDialog(Vaccine.this," Error!!!
\n Aadhar number is invalid");
                        textField.setText("");
                        }
                }}
        );

        textField = new JTextField();
        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(15)
```

```
                                                        .addComponent(lblid)
                                                .addGap(30)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.LEADING, false)
                                                        .addComponent(textField,
GroupLayout.DEFAULT_SIZE, 172, Short.MAX_VALUE))
                                                .addGap(10)

        .addGroup(gl_contentPane.createParallelGroup()
                                                                .addComponent(btnLogin))))
                                        )
                );
                gl_contentPane.setVerticalGroup(
                        gl_contentPane.createParallelGroup(Alignment.LEADING)
                          .addGroup(gl_contentPane.createSequentialGroup()
                                        .addGap(30)

        .addGroup(gl_contentPane.createParallelGroup(Alignment.BASELINE)
                                                .addComponent(lblid)
                                                .addComponent(textField,
GroupLayout.PREFERRED_SIZE, GroupLayout.DEFAULT_SIZE, GroupLayout.PREFERRED_SIZE))
                                        .addGap(20)
                                        .addGroup(gl_contentPane.createParallelGroup()
                                                        .addComponent(btnLogin))
                ));
                contentPane.setLayout(gl_contentPane);
        }
}
```

DB.java:

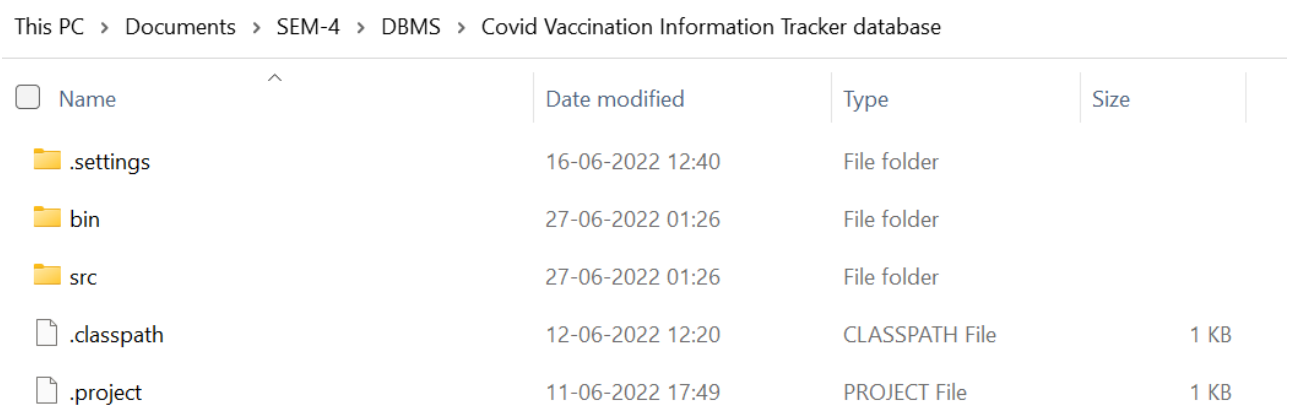
```
import java.sql.Connection;
import java.sql.DriverManager;
public class DB {
        public static Connection getConnection(){
                Connection con=null;
                try{
                        Class.forName("com.mysql.cj.jdbc.Driver");

        con=DriverManager.getConnection("jdbc:mysql://localhost:3306/covid","root","srini
ja");
                }catch(Exception e){System.out.println(e);}
                return con;
        }

}
```

GitHub links and folder structure

https://github.com/Sreenija044/DBMSPROJECT



This PC › Documents › SEM-4 › DBMS › Covid Vaccination Information Tracker database › src

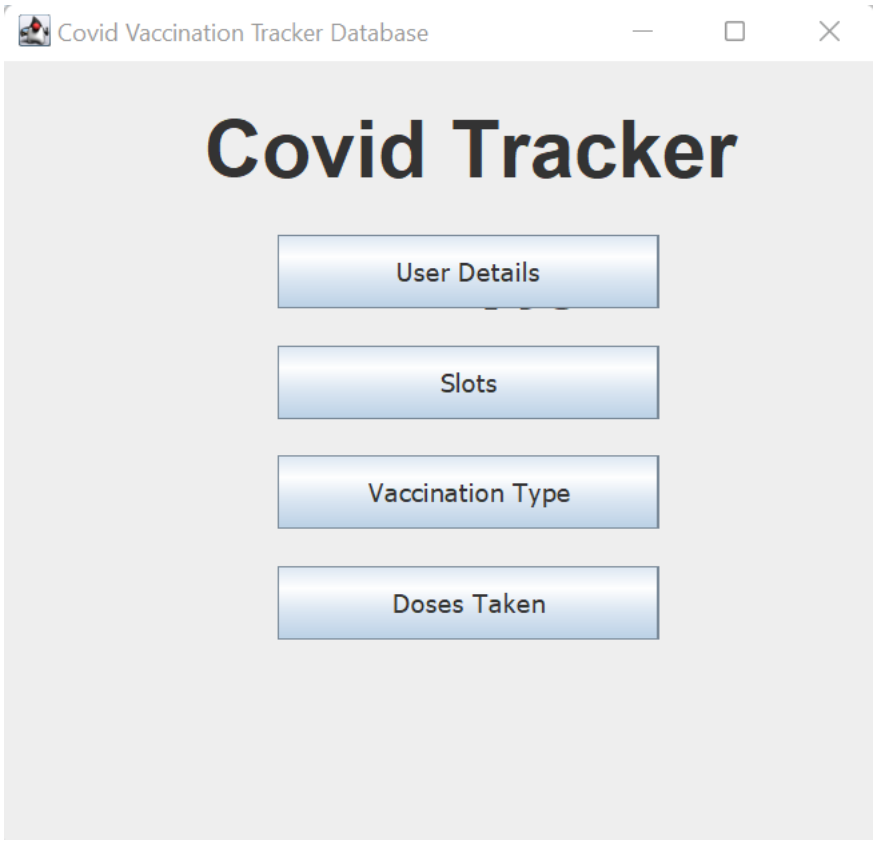
Name	Date modified	Type	Size
AddUser	27-06-2022 00:52	JAVA File	7 KB
checkdoses	27-06-2022 00:14	JAVA File	2 KB
checkvaccine	27-06-2022 00:26	JAVA File	1 KB
checkvaccinetype	27-06-2022 00:23	JAVA File	2 KB
DB	27-06-2022 00:28	JAVA File	1 KB
Dosesdetails	26-06-2022 22:33	JAVA File	2 KB
Dosestaken	27-06-2022 00:29	JAVA File	6 KB
DOSESTAKENUI	12-06-2022 11:56	JAVA File	2 KB
DosesUI	27-06-2022 00:28	JAVA File	4 KB
MainUI	12-06-2022 22:28	JAVA File	3 KB
Menu	27-06-2022 00:22	JAVA File	5 KB
SLOTSUI	27-06-2022 00:34	JAVA File	5 KB
Userdetails	26-06-2022 17:11	JAVA File	2 KB
USERUI	12-06-2022 11:56	JAVA File	2 KB
Vaccine	27-06-2022 00:25	JAVA File	4 KB

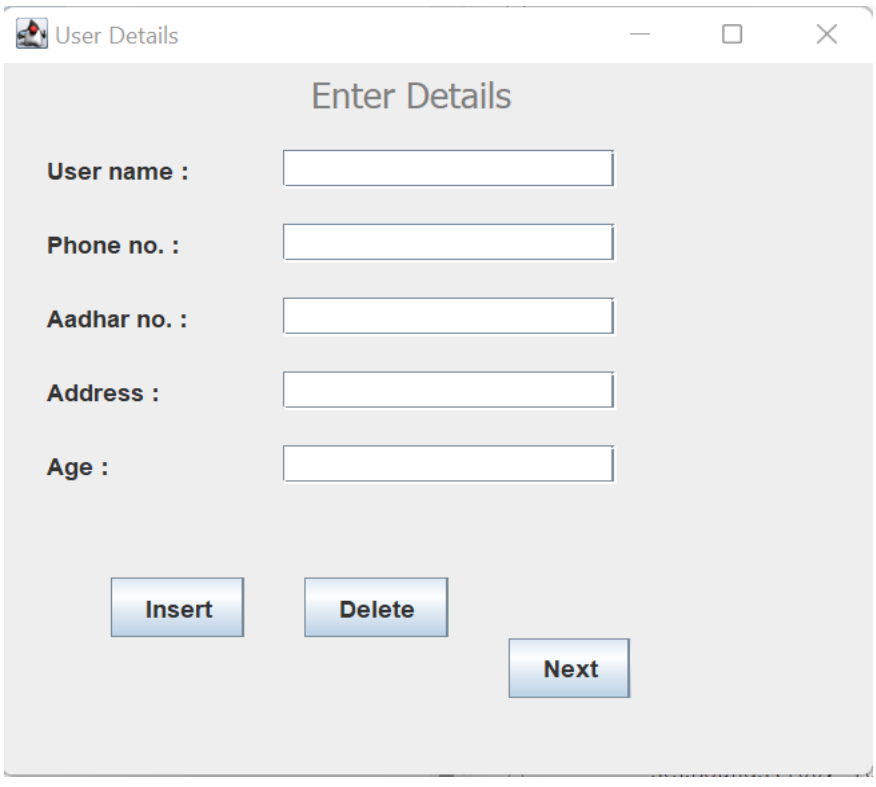
› This PC › Documents › SEM-4 › DBMS › Covid › bin

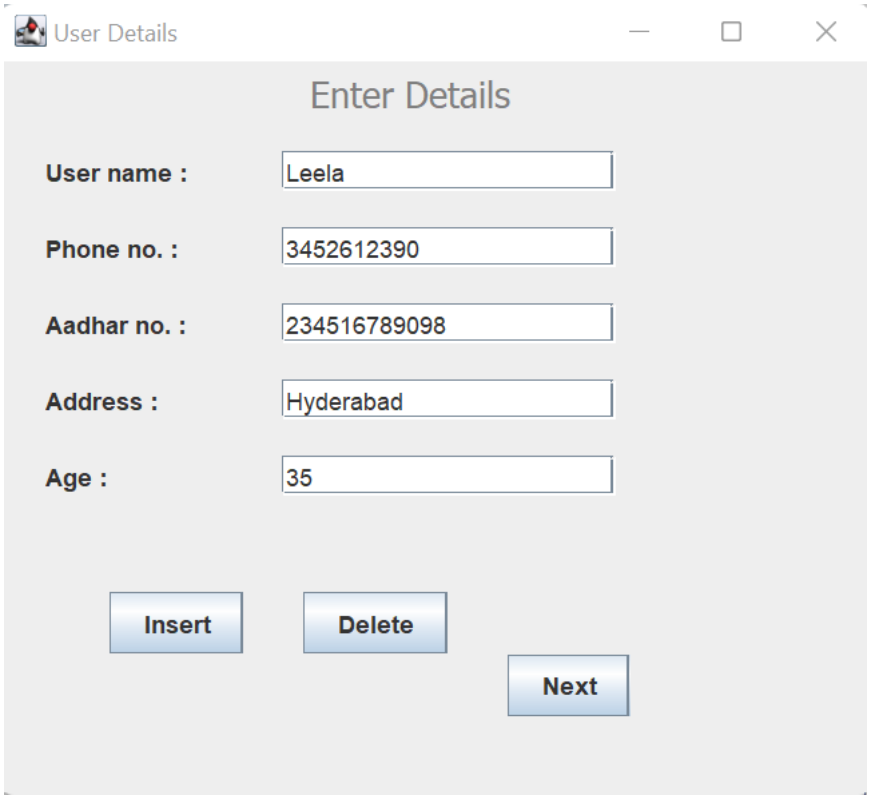
AddUser$1.class	DosesUI$2.class	Vaccine$2.class
AddUser$2.class	DosesUI$3.class	Vaccine$3.class
AddUser$3.class	DosesUI.class	Vaccine.class
AddUser$4.class	Menu$1.class	
AddUser.class	Menu$2.class	
checkdoses.class	Menu$3.class	
checkvaccine.class	Menu$4.class	
checkvaccinetype.class	Menu$5.class	
DB.class	Menu.class	
Dosesdetails.class	SLOTSUI$1.class	
Dosestaken$1.class	SLOTSUI$2.class	
Dosestaken$2.class	SLOTSUI$3.class	
Dosestaken$3.class	SLOTSUI.class	
Dosestaken.class	Userdetails.class	
DosesUI$1.class	Vaccine$1.class	

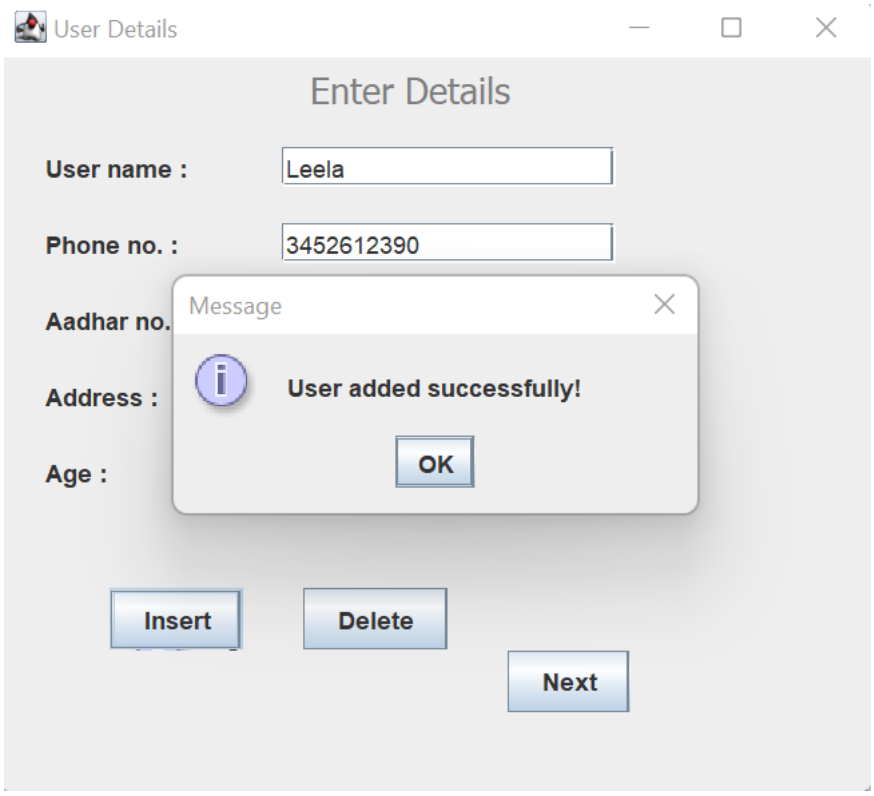
Testing:

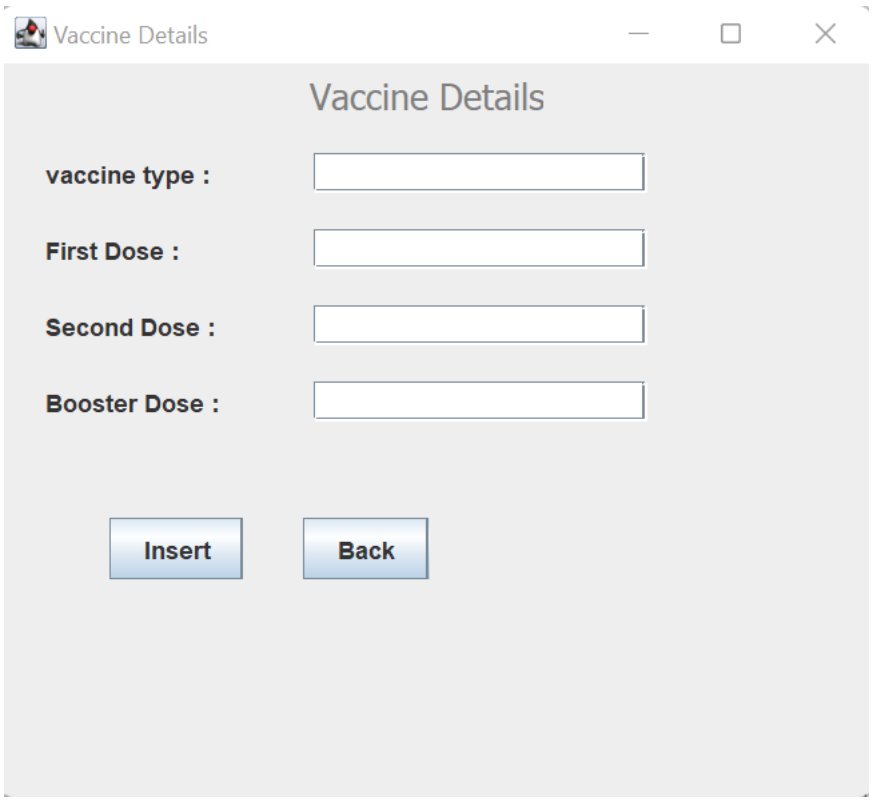
Java GUI Testing:

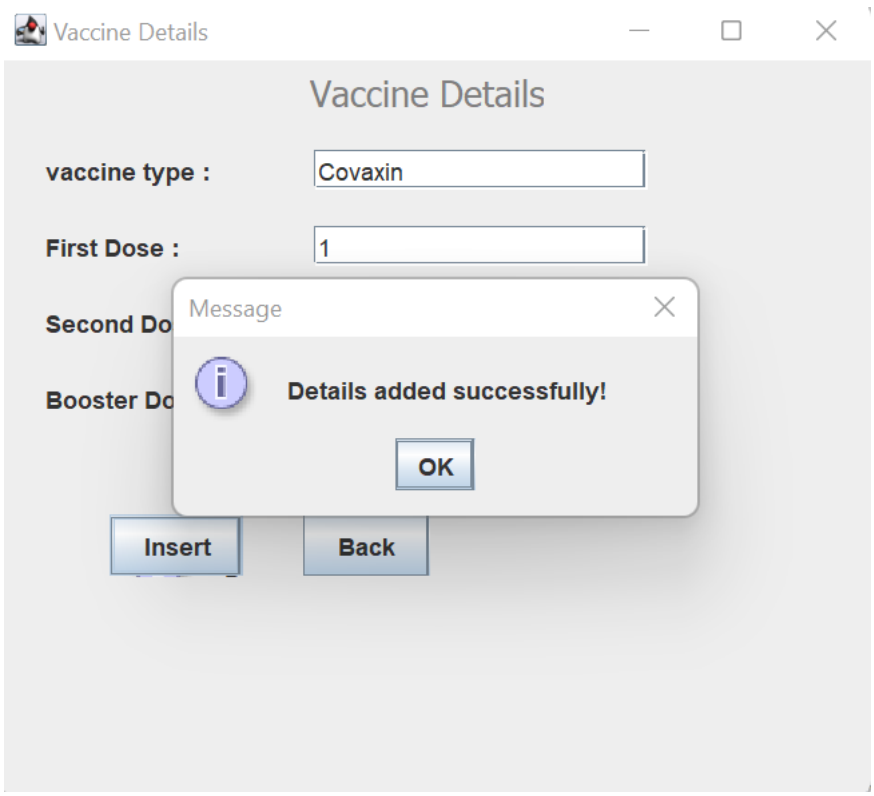


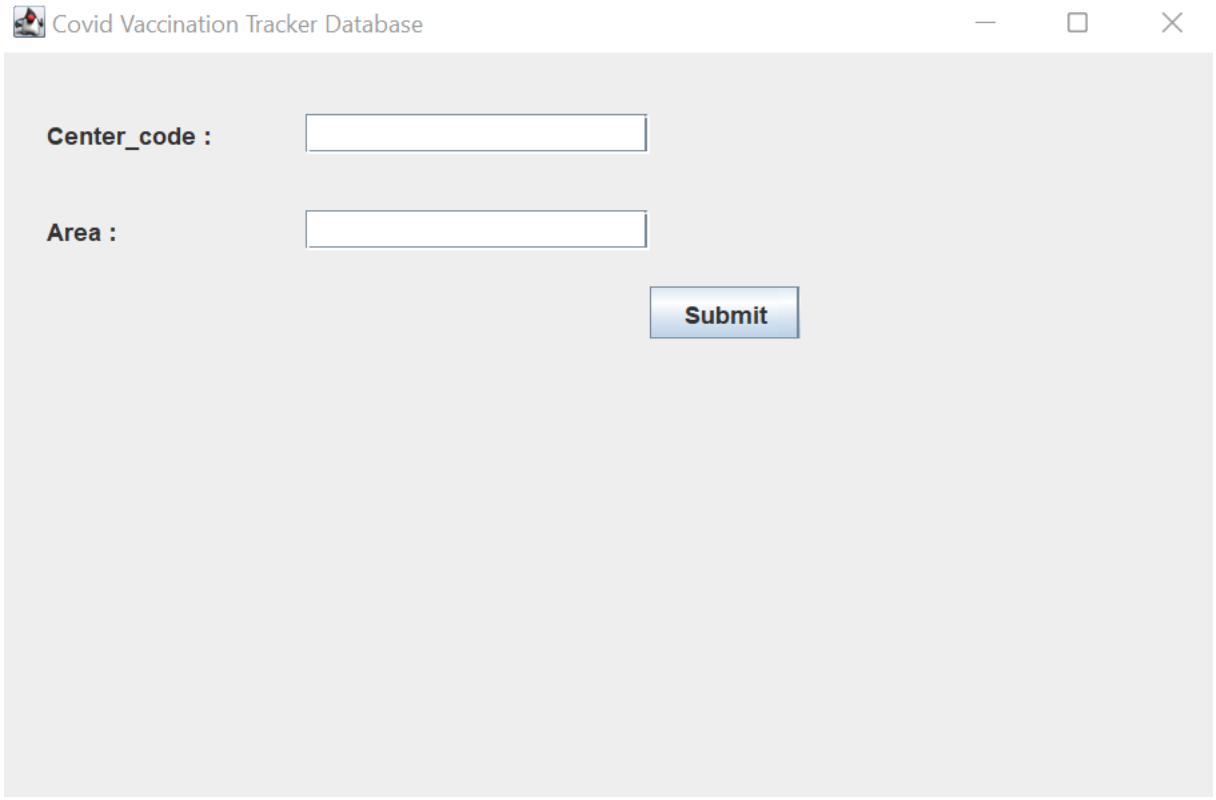


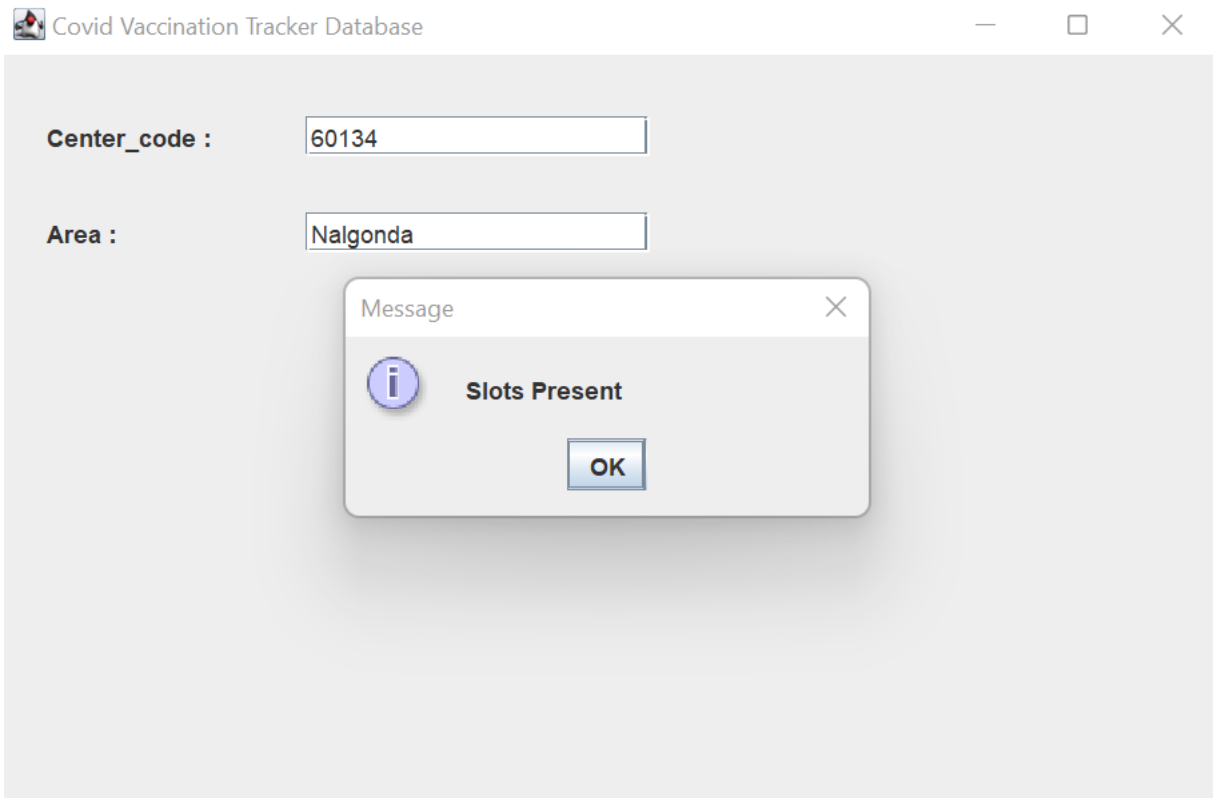

User Details
Enter Details
User name :
Leela
Phone no. :
3452612390
Aadhar no. :
234516789098
Address :
Hyderabad
Age :
35
Insert
Delete
Next

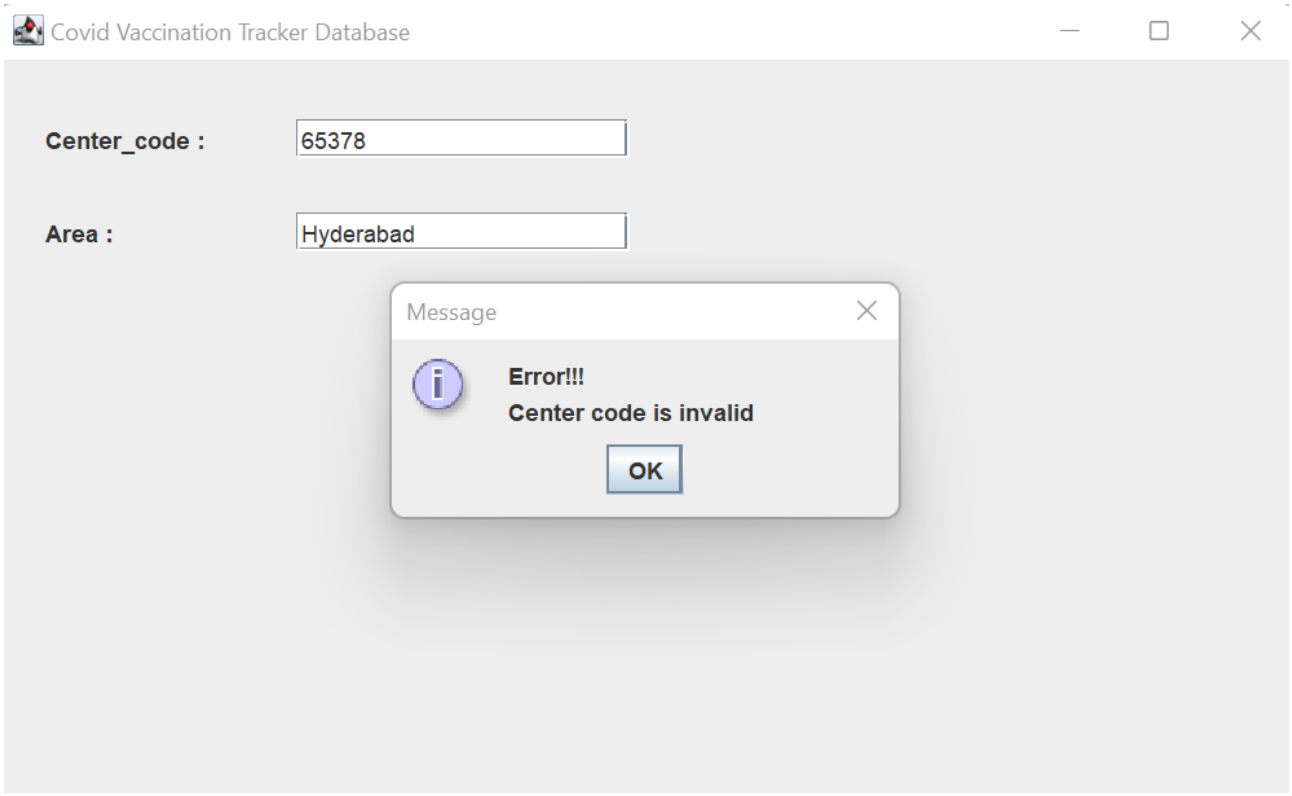

User Details
Enter Details
User name :
Leela
Phone no. :
3452612390
Aadhar no.
Address :
Age :
Message
User added successfully!
OK
Insert
Delete
Next


Vaccine Details
Vaccine Details
vaccine type :
First Dose :
Second Dose :
Booster Dose :
Insert
Back


Vaccine Details
Vaccine Details
vaccine type :
Covaxin
First Dose :
1
Second Do
Booster Do
Message
Details added successfully!
OK
Insert
Back


Covid Vaccination Tracker Database
Center_code :
Area :
Submit


Covid Vaccination Tracker Database
Center_code :
60134
Area :
Nalgonda
Message
Slots Present
OK


Covid Vaccination Tracker Database
Center_code :
65378
Area :
Hyderabad
Message
Error!!!
Center code is invalid
OK

Results:

I successfully completed this PROJECT “Covid Vaccination Information Tracker Database”.

Discussion and Future work

While doing this project I got new ideas I understood how to work on projects. Now to further extend this project I want to create an android app by which I can control my project on my handand connect to it.

References:

- https://www.academia.edu/36893248/Ramakrishnan_-_Database_Management_Systems_3rd_Edition
- https://docs.oracle.com/javase/7/docs/index.html
- https://www.javatpoint.com/dbms-tutorial
- http://www.sqlines.com/articles/java/sql_server_jdbc_connection