

JAVA SWING BASED – CAREER DENDROGRAM – SQL CONNECTIVITY USING JDBC

A Report

*Submitted in partial fulfillment of the
Requirements for the COURSE*

DATABASE MANAGEMENT SYSTEMS LAB

By

Akash<1602-21-737-068>

Bhavani<1602-21-737-076>

Praneetha<1602-21-737-098>

Sreshtha<1602-21-737-115>

Under the guidance of Ms. S. Aruna



**Department of Information
Technology Vasavi College of
Engineering (Autonomous)
(Affiliated to Osmania University)
Ibrahimbagh, Hyderabad-31
2022-2023**

ACKNOWLEDGEMENTS

We are overwhelmed in all humbleness and gratefulness to acknowledge our debt to all those who have helped us put these ideas, well above the level of simplicity and into something concrete. We are extremely thankful to our college, Vasavi College of Engineering, Hyderabad for providing the opportunity to implement our project, "Electronic Medical Records" which helped us in learning many new things. We would like to express our gratitude to Ms. S. Aruna, Associate Professor, Vasavi College of Engineering, for the esteemed guidance, moral support and invaluable advice provided by her for the success of the project. We are also thankful for our friends and peers for their help and encouragement throughout the project.

Akash
1602-21-737-068
Bhavani
1602-21-737-076
Praneetha
1602-21-737-098
Sreshta
1602-21-737-115

Date: 04-07-2023

ABSTRACT

This project explores the implementation and benefits of Electronic Medical Records (EMRs) in healthcare settings. It examines the limitations of paper-based medical records and the advantages of digital EMRs. The project covers technical considerations, such as data storage, interoperability, security, and privacy, as well as different technologies used for EMR creation and management. The benefits of EMRs for healthcare providers, patients, and researchers are discussed, including improved diagnosis, coordinated care, patient empowerment, and research insights. Challenges such as data security, interoperability, user adoption, and ethics are also addressed, along with potential strategies for mitigating them. Overall, the project highlights the transformative potential of EMRs in enhancing healthcare delivery and calls for collaboration among stakeholders to ensure successful implementation.

TABLE OF CONTENTS

1. Requirement analysis
 - 1.1. List of tables
2. Architecture
 - 2.1. Java swings
 - 2.2. SQL
3. Technology
 - 3.1. Software requirements
 - 3.2. Hardware requirements
4. ER Diagram
5. Tables used in database
 - 5.1. Patient table
 - 5.2. Doctor table
 - 5.3. DiagnosticCentre table
 - 5.4. Medical_history
 - 5.5. Hospital
 - 5.6. Medical_insurance
 - 5.7. Patients sign up table
 - 5.8. Doctor sign up table
6. Implementation
 - 6.1. Front end programming and connectivity
 - 6.2. Code
7. Testing
 - 7.1. Outputs
 - 7.1.1. Sign up page
 - 7.1.2. Login page
 - 7.1.3. Patient View
 - 7.1.4. Login output
8. Conclusion
9. Future Scope of the project

1. REQUIREMENT ANALYSIS

1.1. List of tables

- 1.1.1. Patient table
- 1.1.2. Doctor table
- 1.1.3. DiagnosticCentre table
- 1.1.4. Medical_history
- 1.1.5. Hospital
- 1.1.6. Medical_insurance
- 1.1.7. Patients sign up table
- 1.1.8. Doctor sign up table

2. ARCHITECTURE

2.1. JAVA SWINGS

Java swing tutorial is a part of Java Foundation Classes (JFC) that is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Java swing provides platform independent and lightweight components. The javax.swing package provides classes for Java swing API such as JButton, JTextField, JTextArea, JCheckBox, JOptionPane etc. which we have used to perform various functions in our project.

2.2. SQL

Structured 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. We have used Oracle database to implement our project.

3. TECHNOLOGY

3.1. SOFTWARE REQUIREMENTS

The following are the software used in our project –

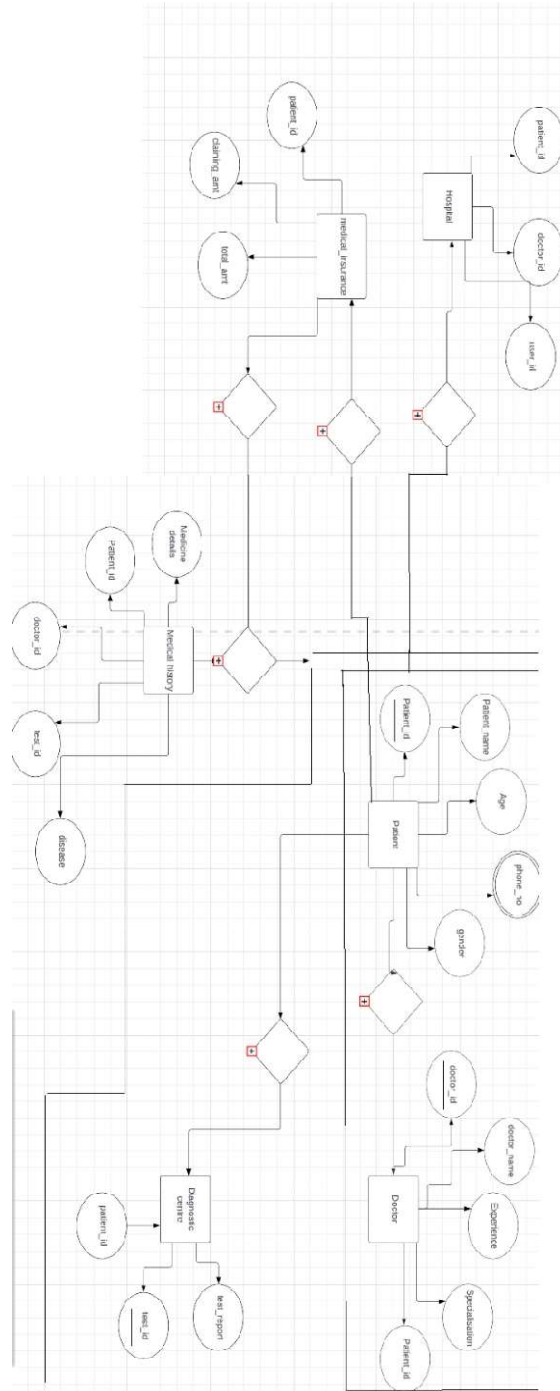
- Java SE version 15.0.2
- Oracle 11g database –
SQLPLUS 11.2.0.2.0

3.2. HARDWARE REQUIREMENTS

The hardware interface requirements ensure the software executes all requests given by the user efficiently. They are –

- Processor – Intel core i5
and above
- RAM – Minimum
2GB/4GB and above

4. ER DIAGRAM



5. TABLES IN DATABASE

5.1. PATIENT TABLE

```
SQL> desc patient;
Name                                     Null?   Type
-----
PATIENT_ID                             NOT NULL NUMBER(20)
PATIENT_NAME                             NOT NULL VARCHAR2(20)
AGE                                     NUMBER(3)
PHONE_NO                                NUMBER(10)
GENDER                                  VARCHAR2(1)
ADDRESS                                 VARCHAR2(30)
```

5.2. DOCTORS TABLE

```
SQL> desc doctor;
Name                                     Null?   Type
-----
DOCTOR_ID                             NOT NULL NUMBER(8)
DOCTOR_NAME                             NOT NULL VARCHAR2(20)
EXP                                     NUMBER(5)
SPL                                     VARCHAR2(20)
```

5.3. DIAGNOSTIC CENTER TABLE

```
SQL> desc diag_centre
Name                                     Null?   Type
-----
TEST_ID                               NOT NULL NUMBER(7)
PATIENT_ID                             NUMBER(20)
REPORT                                 VARCHAR2(50)
READING                                NUMBER(6)
```

5.4. MEDICAL HISTORY

```
SQL> desc medical_history;
Name                                     Null?   Type
-----
PATIENT_ID                             NOT NULL NUMBER(20)
DOCTOR_ID                             NOT NULL NUMBER(8)
DAY                                    NOT NULL DATE
TIME                                  NOT NULL TIMESTAMP(6)
DISEASE                                VARCHAR2(10)
```

5.5. HOSPITAL TABLE

```
SQL> desc hosp;
```

Name	Null?	Type
PATIENT_ID		NUMBER(20)
DOCTOR_ID		NUMBER(8)
HOSP_ID	NOT NULL	NUMBER(6)

5.6. MEDICAL INSURANCE

```
SQL> desc med_ins;
```

Name	Null?	Type
PATIENT_ID		NUMBER(20)
AMT		NUMBER(10)
COMPANY		VARCHAR2(20)
HOSP_ID		NUMBER(6)
DOCTOR_ID		NUMBER(8)

5.7. PATIENT SIGNUP TABLE

```
SQL> desc signpat;
```

Name	Null?	Type
USER_NAME	NOT NULL	VARCHAR2(10)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
PWD		VARCHAR2(10)
CPWD		VARCHAR2(10)
EMAIL		VARCHAR2(20)
PHNO		NUMBER(10)
AGE		NUMBER(3)
GENDER		VARCHAR2(1)
BLOOD		VARCHAR2(4)
ADDRESS		VARCHAR2(20)

5.8. DOCTOR SIGNUP TABLE

```
SQL> desc signdoc;
```

Name	Null?	Type
USER_NAME	NOT NULL	VARCHAR2(11)
FIRST_NAME		VARCHAR2(10)
LAST_NAME		VARCHAR2(10)
PWD		VARCHAR2(10)
CPWD		VARCHAR2(10)
EMAIL		VARCHAR2(20)
AGE		NUMBER(3)
PNO		NUMBER(10)
QUAL		VARCHAR2(20)
SPL		VARCHAR2(10)
EXP		NUMBER(5)
HISTORY		VARCHAR2(20)

6. IMPLEMENTATION

6.1. FRONT END PROGRAMMING AND CONNECTIVITY

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 a part of the Java 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 –

6.2. CODE

```
/*
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/
 Licenses/license-default.txt to change this
 license
 * Click
 nbfs://nbhost/SystemFileSystem/Templates/
 GUIForms/JFrame.java to edit this template
 */

/**
 *
 * @author prane
 */
import java.sql.Connection;
import java.sql.DriverManager;
```

```

import java.sql.ResultSet;
import java.sql.Statement;
import javax.swing.JOptionPane;
public class pat extends javax.swing.JFrame {

    /**
     * Creates new form pat
     */
    public pat() {
        initComponents();
    }

    /**
     * This method is called from within the
     constructor to initialize the form.
     * WARNING: Do NOT modify this code.
     The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed"
    desc="Generated Code">
    private void initComponents() {

        jToggleButton1 = new
javax.swing.JToggleButton();
        jToggleButton2 = new
javax.swing.JToggleButton();
        jLabel2 = new javax.swing.JLabel();
        id1 = new javax.swing.JTextField();
        jLabel3 = new javax.swing.JLabel();

        setDefaultCloseOperation(javax.swing.Wind
owConstants.EXIT_ON_CLOSE);
        getContentPane().setLayout(new
org.netbeans.lib.awtextra.AbsoluteLayout());

```

```

        jToggleButton1.setText("VIEW");
        jToggleButton1.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent
evt) {

jToggleButton1ActionPerformed(evt);
            }
        });
        getContentPane().add(jToggleButton1,
new
org.netbeans.lib.awtextra.AbsoluteConstrain
ts(50, 240, 100, 50));

        jToggleButton2.setText("MODIFY");
        getContentPane().add(jToggleButton2,
new
org.netbeans.lib.awtextra.AbsoluteConstrain
ts(300, 240, 111, 50));

        jLabel2.setFont(new
java.awt.Font("Segoe UI", 0, 18)); // NOI18N
        jLabel2.setForeground(new
java.awt.Color(255, 255, 255));
        jLabel2.setText("Enter your Patient ID");
        getContentPane().add(jLabel2, new
org.netbeans.lib.awtextra.AbsoluteConstrain
ts(160, 70, 190, 50));

        id1.addActionListener(new
java.awt.event.ActionListener() {
            public void
actionPerformed(java.awt.event.ActionEvent
evt) {

            id1ActionPerformed(evt);

```

```

    }
    });
    getContentPane().add(id1, new
org.netbeans.lib.awtextra.AbsoluteConstrain
ts(160, 150, 170, 20));

    jLabel3.setIcon(new
javax.swing.ImageIcon("C:\\Users\\prane\\D
ownloads\\Screenshot (50).png")); //
NOI18N
    getContentPane().add(jLabel3, new
org.netbeans.lib.awtextra.AbsoluteConstrain
ts(0, 0, 950, -1));

    pack();
} // </editor-fold>

```

```

private void
jToggleButton1ActionPerformed(java.awt.ev
ent.ActionEvent evt) {
    // TODO add your handling code here
    try {
        Connection con =
        JavaApplication2.getCon();
        Statement stmt =
        con.createStatement();

```

```

        int id = Integer.parseInt(id1.getText()); //
        Assuming you have a text field named "id1"
        for entering patient ID

```

```

        // Fetching patient details
        String query = "SELECT patient_name,
age, report, reading, address, phone_no
FROM patient JOIN diag_centre ON
patient.patient_id = diag_centre.patient_id
WHERE patient.patient_id = " + id;

```

```

        String k="select report from diag_centre
where patient_id="+id;
        ResultSet resultSet =
stmt.executeQuery(query);
        ResultSet w =
stmt.executeQuery(query);
        System.out.println(w);
        if (resultSet.next()) {
            String name =
resultSet.getString("patient_name");
            int age = resultSet.getInt("age");
            String report =
resultSet.getString("report");
            String reading =
resultSet.getString("reading");
            String address =
resultSet.getString("address");
            int pno =
resultSet.getInt("phone_no");

```

```

JOptionPane.showMessageDialog(null,
"Welcome!\nYour details are:\nPatient
Name: " + name
        + "\nAge: " + age + "\nReport
details: " + report + "\nReadings: " + reading
        + "\nAddress: " + address +
"\nPhone number: " + pno);
    } else {

```

```

JOptionPane.showMessageDialog(null,
"Patient details not found.");
    }

```

```

        resultSet.close();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null,

```

```
e);  
}  
}
```

```
private void  
id1ActionPerformed(java.awt.event.ActionEv  
ent evt) {  
    // TODO add your handling code here:  
}
```

```
/**  
 * @param args the command line  
arguments  
 */  
public static void main(String args[]) {  
    /* Set the Nimbus look and feel */  
    //<editor-fold defaultstate="collapsed"  
desc=" Look and feel setting code (optional)  
>  
    /* If Nimbus (introduced in Java SE 6) is  
not available, stay with the default look and  
feel.  
    * For details see  
http://download.oracle.com/javase/tutorial/  
uiswing/lookandfeel/plaf.html  
    */  
    try {  
        for  
(javax.swing.UIManager.LookAndFeelInfo  
info :  
javax.swing.UIManager.getInstalledLookAnd  
Feels()) {  
            if  
("Nimbus".equals(info.getName())) {  
  
javax.swing.UIManager.setLookAndFeel(info.  
getClassName());
```



```

        break;
    }
}
} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(pat.class.g
etName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(pat.class.g
etName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(pat.class.g
etName()).log(java.util.logging.Level.SEVERE,
null, ex);
    } catch
(javax.swing.UnsupportedLookAndFeelExcep
tion ex) {

java.util.logging.Logger.getLogger(pat.class.g
etName()).log(java.util.logging.Level.SEVERE,
null, ex);
    }
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new
Runnable() {
    public void run() {
        new pat().setVisible(true);
    }
});
}

```

```
// Variables declaration - do not modify
private javax.swing.JTextField id1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JToggleButton
jToggleButton1;
private javax.swing.JToggleButton
jToggleButton2;
// End of variables declaration
}
```

7. TESTING

7.1. OUTPUTS

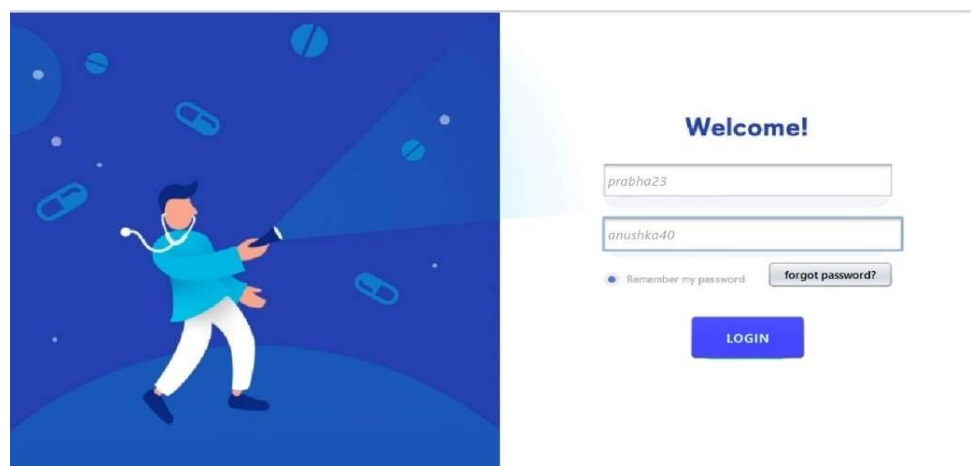
7.1.1. SIGN UP PAGE



A screenshot of a 'SIGN UP' form on a dark blue background with a medical theme. On the left is a large caduceus symbol. The form fields on the right are: Username (praneet45), First Name (Praneetha), Last Name (Balanaga), Password (praneetha), Confirm Password (praneetha), Email (praneet@gmail.com), Phone number (8819841286), Age (18), Gender (F), Blood Group (O), and Address (Hyderabad). A 'Sign Up' button is at the bottom right.

Field	Value
Username	praneet45
First Name	Praneetha
Last Name	Balanaga
Password	praneetha
Confirm Password	praneetha
Email	praneet@gmail.com
Phone number	8819841286
Age	18
Gender	F
Blood Group	O
Address	Hyderabad

7.1.2. LOGIN PAGE



A screenshot of a 'Welcome!' login page. The left side features an illustration of a doctor in a blue coat with a stethoscope, holding a flashlight. The right side contains the login form with fields for Username (prabhaz3) and Password (anushka40). Below the password field is a 'Remember my password' checkbox (checked) and a 'forgot password?' link. A blue 'LOGIN' button is at the bottom.

Field	Value
Username	prabhaz3
Password	anushka40

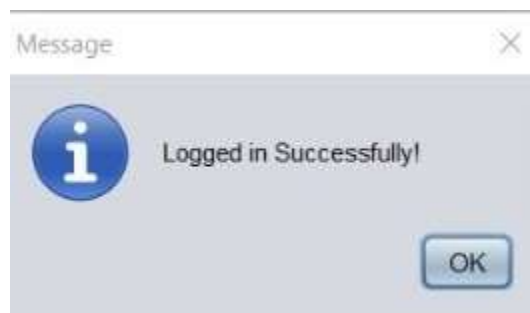
☒ Remember my password [forgot password?](#)

LOGIN

7.1.3. PATIENT VIEW



7.1.4. LOGIN OUTPUT



8. CONCLUSION

The project "Electronic Medical Records" has emphasized the significant benefits of implementing EMRs in healthcare, including improved accessibility, efficiency, and accuracy of patient information. The project has explored technical considerations, such as data storage, interoperability, security, and privacy, while also addressing the potential challenges and concerns associated with EMRs.

The project has highlighted the wide-ranging advantages of EMRs, including enhanced patient care, coordinated healthcare delivery, patient empowerment, and valuable insights for research and population health management. It has also identified areas for future development, such as interoperability, artificial intelligence, patient engagement, telemedicine, blockchain technology, and integration with IoT devices.

Overall, the project underscores the need for collaboration among healthcare organizations, policymakers, and technology stakeholders to fully realize the transformative potential of EMRs. By embracing digital transformation and effectively implementing EMRs, the healthcare industry can achieve more efficient, patient-centric, and data-driven care, ultimately improving health outcomes for individuals and society.

9. FUTURE SCOPE OF THE PROJECT

The future scope of the project "Electronic Medical Records" includes enhancing interoperability and standardization, integrating artificial intelligence and data analytics, empowering patients through personal health records, incorporating telemedicine and remote monitoring, exploring blockchain technology for data security, addressing ethical and legal considerations, integrating with Internet of Things devices, and leveraging EMRs for population health management. These advancements aim to improve data exchange, clinical decision-making, patient engagement, remote care, data privacy, and public health initiatives. The project contributes to the ongoing development of EMRs for more efficient and patient-centric healthcare systems.