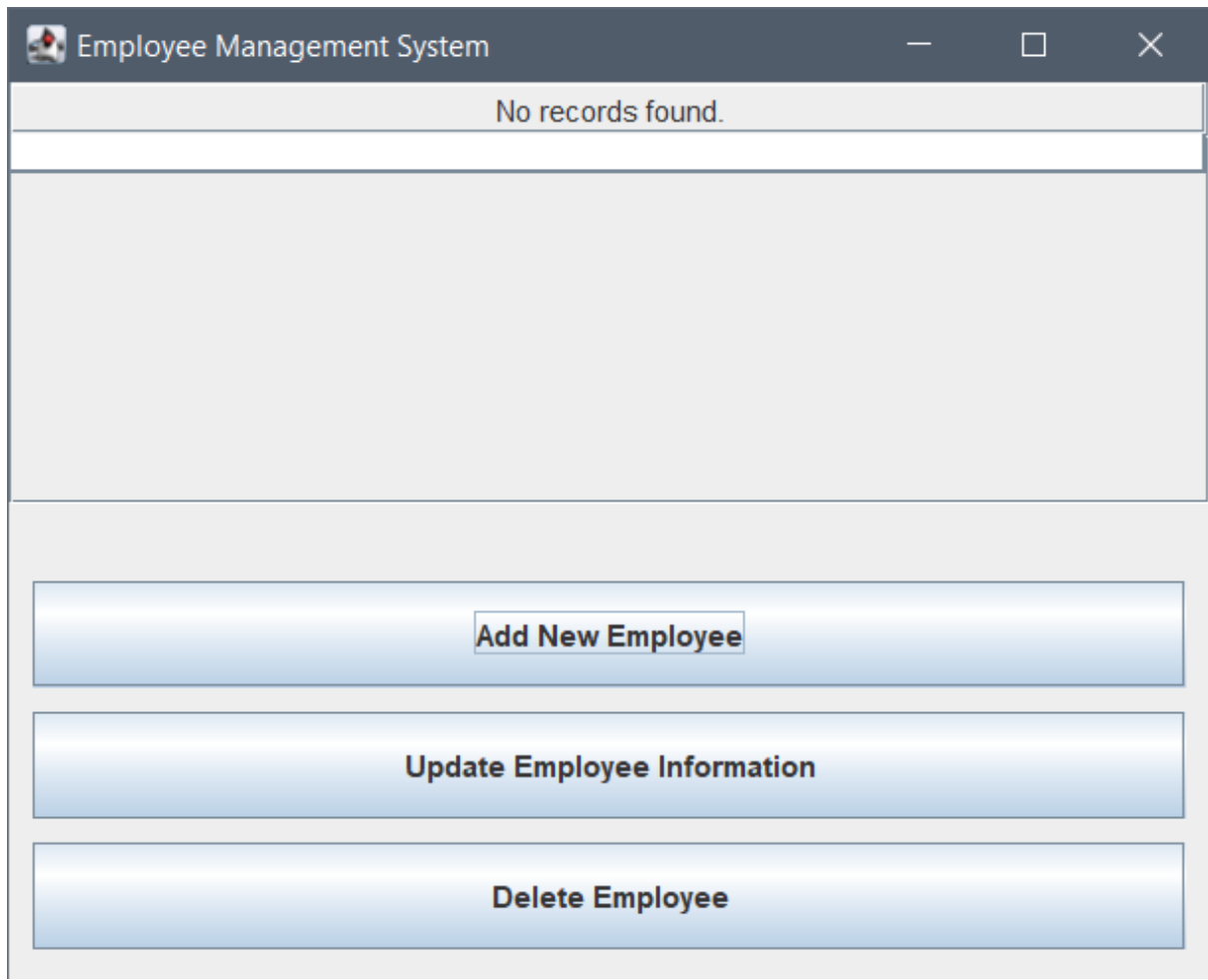


DIGITAL ASSIGNMENT 1


NAME: Kamran Ansari

REG NO: 22MCA0223

Output



The screenshot displays a window titled "Employee Management System" with standard window controls (minimize, maximize, close). The main content area shows a message "No records found." above a large, empty light gray rectangular box. Below this, there are three blue buttons with white text, stacked vertically: "Add New Employee", "Update Employee Information", and "Delete Employee".

 Add New Employee

Employee ID:

2201

Employee Name:

Jake

Employee Email:

jake@email.com

Add

Add New Employee

Update Employee Information

Delete Employee



Employee Management System



Record #	ID	Name	Email
1	2201	Jake	jake@email.com

Add New Employee

Update Employee Information

Delete Employee



Employee Management System



Record #	ID	Name	Email
1	2201	Jake	jake@email.com
2	2202	Jay	jay@email.com
3	2204	Jim	jim@email.com
4	2203	Henry	henry@email.com

Add New Employee

Update Employee Information

Delete Employee

Update Employee In...

— □ ×

Employee ID:

2201

Employee Name:

Jake Ryan

Employee Email:

jakeryan@email.com


Update

Name	Email
	jake@email.com
	jay@email.com
	jim@email.com
	henry@email.com

Add New Employee


Update Employee Information

Delete Employee

Employee Management System

Record #	ID	Name	Email
1	2201	Jake	jake@email.com
2	2202	Jay	jay@email.com
3	2204	Jim	jim@email.com
4	2203	Henry	henry@email.com

Success

Record updated successfully.

OK

Add New Employee

Update Employee Information

Delete Employee



Employee Management System



Record #	ID	Name	Email
1	2202	Jay	jay@email.com
2	2204	Jim	jim@email.com
3	2203	Henry	henry@email.com
4	2201	Jake Ryan	jakeryan@emai...

Add New Employee

Update Employee Information

Delete Employee

D...

—

□

×

Employee ID:

Delete

		Name	Email
		Jay	jay@email.com
		Jim	jim@email.com
		Henry	henry@email.com
4	2201	Jake Ryan	jakeryan@emai...

Add New Employee

Update Employee Information

Delete Employee



Employee Management System



Record #	ID	Name	Email
1	2202	Jay	jay@email.com
2	2204	Jim	jim@email.com
3	2203	Henry	henry@email.com
4	2201	Jake Ryan	jakeryan@emai...

Error



Record not found.

OK

Update Employee Information

Delete Employee

D...

—

□

×

Employee ID:

Delete

	Name	Email
	Jay	jay@email.com
	Jim	jim@email.com
	Henry	henry@email.com
4	Jake Ryan	jakeryan@emai...

Add New Employee

Update Employee Information

Delete Employee



Employee Management System



Record #	ID	Name	Email
1	2202	Jay	jay@email.com
2	2204	Jim	jim@email.com
3	2203	Henry	henry@email.com
4	2201	Jake Ryan	jakeryan@emai...

Success




Record deleted successfully.

OK

Update Employee Information

Delete Employee

 Employee Management System

—

□

×

Record #	ID	Name	Email
1	2202	Jay	jay@email.com
2	2204	Jim	jim@email.com
3	2203	Henry	henry@email.com

Add New Employee

Update Employee Information

Delete Employee

Code

```
import java.awt.*;
import java.awt.event.*;
import java.sql.*;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;

public class CRUDApplication extends JFrame {
    private JButton createButton, updateButton, deleteButton;
    private Connection connection;
    private JTable resultTable;
    private final String DATABASE_NAME = "employee";
    private final String DATABASE_USERNAME = "postgres";
    private final String DATABASE_PASSWORD = "1234";
    private final String DATABASE_TABLE = "employees";

    public CRUDApplication() {
        setTitle("Employee Management System");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        createButton = new JButton("Add New Employee");
        updateButton = new JButton("Update Employee Information");
        deleteButton = new JButton("Delete Employee");

        resultTable = new JTable();

        createButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                openCreateWindow();
            }
        });
    }
}
```

```

        }
    });

    updateButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            openUpdateWindow();
        }
    });

    deleteButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            openDeleteWindow();
        }
    });

    setLayout(new GridLayout(2, 1, 20, 20));

    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new GridLayout(3, 1, 10, 10));
    buttonPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));
    buttonPanel.add(createButton);
    buttonPanel.add(updateButton);
    buttonPanel.add(deleteButton);

    add(new JScrollPane(resultTable));
    add(buttonPanel);

    try {
        Class.forName("org.postgresql.Driver");
        String url = "jdbc:postgresql://localhost:5432/" +
DATABASE_NAME;

```

```

        connection = DriverManager.getConnection(url,
DATABASE_USERNAME, DATABASE_PASSWORD);
    } catch (Exception ex) {
        System.out.println(ex);
        JOptionPane.showMessageDialog(this, "Failed to connect to the
database.", "Error",
            JOptionPane.ERROR_MESSAGE);
        System.exit(1);
    }

    readRecords();
}

```

```

private void openCreateWindow() {
    JFrame createFrame = new JFrame("Add New Employee");
    JPanel createPanel = new JPanel();
    createFrame.setSize(300, 200);
    createPanel.setSize(300, 200);
    createPanel.setLayout(new GridLayout(4, 2, 5, 10));
    createPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));
}

```

```

JLabel idLabel = new JLabel("Employee ID:");
JTextField idField = new JTextField(10);
JLabel nameLabel = new JLabel("Employee Name:");
JTextField nameField = new JTextField(20);
JLabel emailLabel = new JLabel("Employee Email:");
JTextField emailField = new JTextField(20);
JButton addButton = new JButton("Add");

```

```

addButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

```

```

        createRecord(Integer.valueOf(idField.getText()),
nameField.getText(), emailField.getText());
        createFrame.dispose();
    }
});

```

```

createPanel.add(idLabel);
createPanel.add(idField);
createPanel.add(nameLabel);
createPanel.add(nameField);
createPanel.add(emailLabel);
createPanel.add(emailField);
createPanel.add(addButton);

```

```

createFrame.add(createPanel);

```

```

createPanel.setVisible(true);
createFrame.setVisible(true);

```

```

}

```

```

private void createRecord(int id, String name, String email) {
    try {
        PreparedStatement statement = connection
            .prepareStatement("INSERT INTO " + DATABASE_TABLE + "
(id, name, email) VALUES (?, ?, ?)");
        statement.setInt(1, id);
        statement.setString(2, name);
        statement.setString(3, email);
        statement.executeUpdate();

        JOptionPane.showMessageDialog(this, "Record created
successfully.", "Success",
            JOptionPane.INFORMATION_MESSAGE);
    }
}

```



```

        } catch (SQLException ex) {
            System.out.println(ex);

            JOptionPane.showMessageDialog(this, "Failed to create the
record.", "Error", JOptionPane.ERROR_MESSAGE);
        }

        readRecords();
    }

    private void openUpdateWindow() {
        JFrame updateFrame = new JFrame("Update Employee Information");
        JPanel updatePanel = new JPanel();
        updatePanel.setSize(300, 200);
        updateFrame.setSize(300, 200);
        updatePanel.setLayout(new GridLayout(4, 2, 5, 10));
        updatePanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JLabel idLabel = new JLabel("Employee ID:");
        JTextField idField = new JTextField(10);
        JLabel nameLabel = new JLabel("Employee Name:");
        JTextField nameField = new JTextField(20);
        JLabel emailLabel = new JLabel("Employee Email:");
        JTextField emailField = new JTextField(20);
        JButton updateButton = new JButton("Update");

        updateButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                updateRecord(Integer.valueOf(idField.getText()),
nameField.getText(), emailField.getText());
                updateFrame.dispose();
            }
        });
    }

```

```

        updatePanel.add(idLabel);
        updatePanel.add(idField);
        updatePanel.add(nameLabel);
        updatePanel.add(nameField);
        updatePanel.add(emailLabel);
        updatePanel.add(emailField);
        updatePanel.add(updateButton);

        updateFrame.add(updatePanel);

        updatePanel.setVisible(true);
        updateFrame.setVisible(true);
    }

    private void updateRecord(int id, String name, String email) {
        try {
            PreparedStatement statement = connection
                .prepareStatement("UPDATE " + DATABASE_TABLE + " SET
name = ?, email = ? WHERE id = ?");
            statement.setString(1, name);
            statement.setString(2, email);
            statement.setInt(3, id);
            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Record updated
successfully.", "Success",
                    JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Record not found.",
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    }

```

```

        } catch (SQLException ex) {
            System.out.println(ex);

            JOptionPane.showMessageDialog(this, "Failed to update the
record.", "Error", JOptionPane.ERROR_MESSAGE);
        }

        readRecords();
    }

    private void openDeleteWindow() {
        JFrame deleteFrame = new JFrame("Delete Employee");
        JPanel deletePanel = new JPanel();
        deletePanel.setSize(200, 100);
        deleteFrame.setSize(200, 100);
        deletePanel.setLayout(new GridLayout(2, 2, 5, 10));
        deletePanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10,
10));

        JLabel idLabel = new JLabel("Employee ID:");
        JTextField idField = new JTextField(10);
        JButton deleteButton = new JButton("Delete");

        deleteButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                deleteRecord(Integer.valueOf(idField.getText()));
                deleteFrame.dispose();
            }
        });

        deletePanel.add(idLabel);
        deletePanel.add(idField);
        deletePanel.add(new JLabel());
        deletePanel.add(deleteButton);
    }

```

```

        deleteFrame.add(deletePanel);

        deletePanel.setVisible(true);
        deleteFrame.setVisible(true);
    }

    private void deleteRecord(int id) {
        try {
            PreparedStatement statement = connection
                .prepareStatement("DELETE FROM " + DATABASE_TABLE + "
WHERE id = ?");
            statement.setInt(1, id);
            int rowsAffected = statement.executeUpdate();

            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Record deleted
successfully.", "Success",
                    JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Record not found.",
"Error", JOptionPane.ERROR_MESSAGE);
            }
        } catch (SQLException ex) {
            System.out.println(ex);
            JOptionPane.showMessageDialog(this, "Failed to delete the
record.", "Error", JOptionPane.ERROR_MESSAGE);
        }

        readRecords();
    }

    private void readRecords() {

```

```

try {
    Statement statement = connection.createStatement();
    ResultSet resultSet = statement.executeQuery("SELECT * FROM "
+ DATABASE_TABLE);

    DefaultTableModel tableModel = new DefaultTableModel();
    tableModel.addColumn("Record #");
    tableModel.addColumn("ID");
    tableModel.addColumn("Name");
    tableModel.addColumn("Email");

    int i = 0;
    while (resultSet.next()) {
        int id = resultSet.getInt("id");
        String name = resultSet.getString("name");
        String email = resultSet.getString("email");

        tableModel.addRow(new Object[] { (i + 1), id, name, email
});

        i++;
    }

    if (i == 0) {
        resultTable.setModel(new DefaultTableModel(new Object[] {
"No records found." }, 1));
    } else {
        resultTable.setModel(tableModel);
    }
} catch (SQLException ex) {
    System.out.println(ex);

    JOptionPane.showMessageDialog(this, "Failed to retrieve
records.", "Error", JOptionPane.ERROR_MESSAGE);
}

```

```
}
```

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

```
    new CRUDApplication().setVisible(true);
```

```
}
```

```
}
```