

Advanced Programming

JAVA SWING DOCUMENTATION
ANTHONY BOMPETSHI

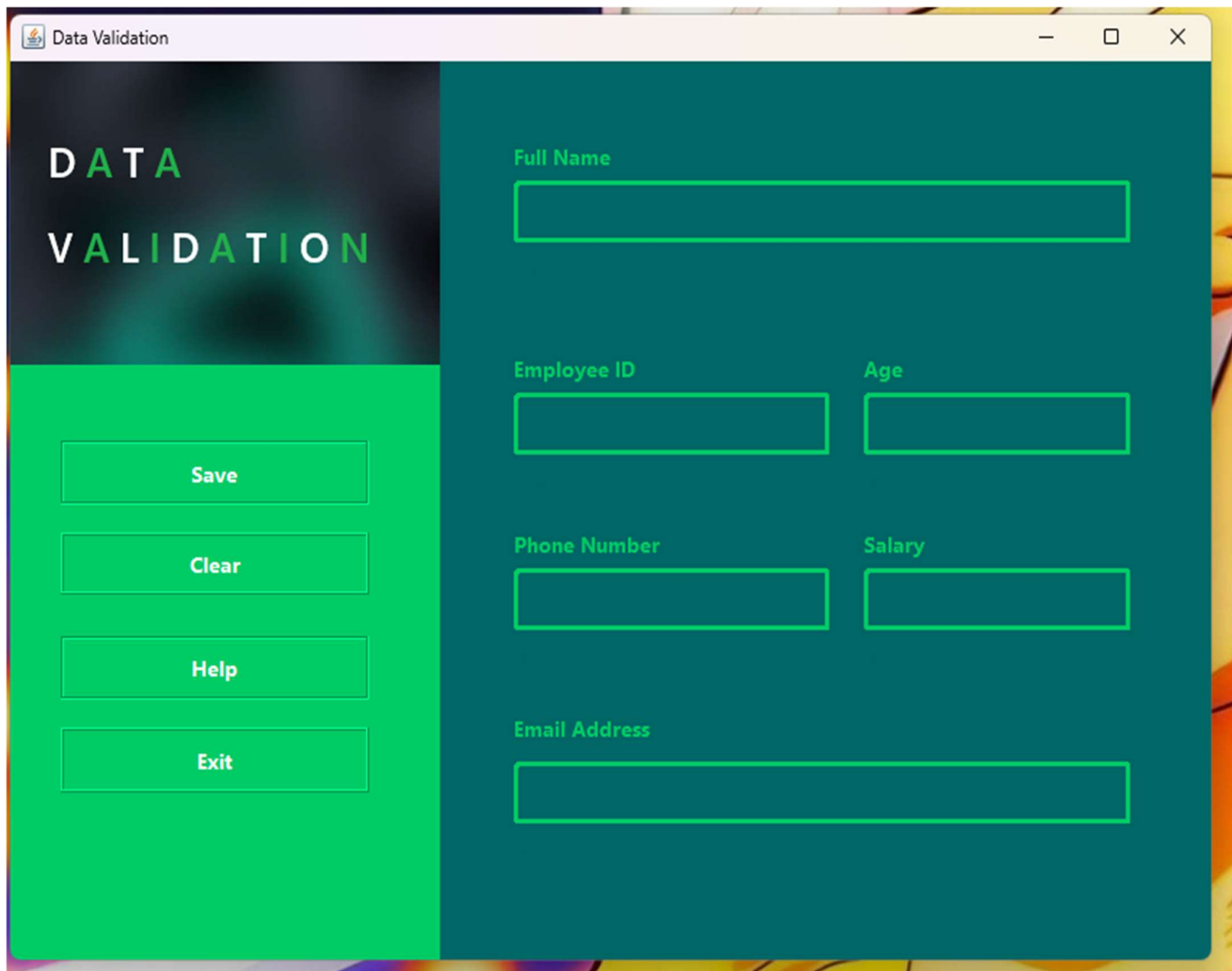
Table of Contents

Cover Page	0
Table Of Contents.....	1
Fields.....	2
Before Testing.....	3
After Testing	4
If All Input Is Valid.....	5
Presence Check.....	6
Type Check.....	7
Length Check.....	8
Code.....	9

Fields

<u>Field Name</u>	<u>Input Type</u>	<u>Input Component</u>	<u>Validation Rules</u>	<u>Reason for Validation</u>	<u>Error Message</u>
<u>Full Name</u>	String	Text Field	<ul style="list-style-type: none"> • Presence Check • Type Check • Length Check (2-50 chars) 	<ul style="list-style-type: none"> • Name cannot be empty • Should contain only letters, spaces, hyphens, apostrophes • Must be between 2-50 characters 	"Full Name is required" "Use only letters, spaces, hyphens (2-50 chars)"
<u>Employee ID</u>	String	Text Field	<ul style="list-style-type: none"> • Presence Check • Format Check • Length Check (8 chars) 	<ul style="list-style-type: none"> • ID cannot be empty • Must follow EMP##### format • Must be exactly 8 characters 	"Employee ID is required" "Format must be EMP12345"
<u>Age</u>	Integer	Text Field	<ul style="list-style-type: none"> • Presence Check • Type Check • Range Check (18-65) 	<ul style="list-style-type: none"> • Age cannot be empty • Must be numerical value • Must be between 18-65 years 	"Age is required" "Age must be a number" "Age must be between 18-65"
<u>Salary</u>	Double	Text Field	<ul style="list-style-type: none"> • Presence Check • Type Check • Range Check (15,000-500,000) 	<ul style="list-style-type: none"> • Salary cannot be empty • Must be numerical value • Must be between 15,000-500,000 	"Salary is required" "Salary must be a number" "Salary must be 15,000-500,000"
<u>Phone Number</u>	String	Text Field	<ul style="list-style-type: none"> • Presence Check • Format Check • Length Check (10 digits) 	<ul style="list-style-type: none"> • Phone number cannot be empty • Must start with 0 and contain only digits • Must be exactly 10 digits 	"Phone number is required" "Must be 10 digits starting with 0"
<u>Email Address</u>	String	Text Field	<ul style="list-style-type: none"> • Presence Check • Format Check 	<ul style="list-style-type: none"> • Email cannot be empty • Must follow valid email format (user@domain.com) 	"Email is required" "Invalid email format"

Before Testing



The screenshot shows a window titled "Data Validation" with a standard Windows title bar (minimize, maximize, close buttons). The window is divided into two main sections. On the left, there is a dark blue header area with the text "DATA" in white and "VALIDATION" in green. Below this header is a light blue sidebar containing four buttons: "Save", "Clear", "Help", and "Exit". On the right, the main content area has a light blue background and contains five input fields with labels above them: "Full Name", "Employee ID", "Age", "Phone Number", and "Salary". The "Email Address" label is positioned above a long input field at the bottom. All input fields are empty.

DATA VALIDATION	
Full Name <input type="text"/>	
Employee ID <input type="text"/>	Age <input type="text"/>
Phone Number <input type="text"/>	Salary <input type="text"/>
Email Address <input type="text"/>	

After Testing

The screenshot shows a window titled "Data Validation" with a dark blue header and a light blue sidebar. The sidebar contains four buttons: "Save", "Clear", "Help", and "Exit". The main area has a teal background and contains several form fields with labels and validation messages.

Field Label	Input Value	Validation Message
Full Name	dfitty	✓ Valid
Employee ID		Employee ID is required
Age	30	✓ Valid
Phone Number	0978768766	✓ Valid
Salary		Salary is required
Email Address		Email Address is required

All inputs is Valid

The screenshot shows a window titled "Data Validation" with a dark blue header and a light blue sidebar. The sidebar contains four buttons: "Save", "Clear", "Help", and "Exit". The main area has a dark blue background with white text for labels and input fields. All input fields are highlighted with a red border, and each has a green checkmark and the word "Valid" below it.

Field	Value	Status
Full Name	Anthony Bompetsi	✓ Valid
Employee ID	EMP12345	✓ Valid
Age	30	✓ Valid
Phone Number	0978768766	✓ Valid
Salary	15000	✓ Valid
Email Address	user@gmail.com	✓ Valid

Precence Check

The screenshot shows a window titled "Data Validation" with a dark teal header and a light teal body. On the left, a dark sidebar contains the text "DATA VALIDATION" and four buttons: "Save", "Clear", "Help", and "Exit". The main area contains five form fields, each with a red error message below it:

- Full Name**: Full Name is required
- Employee ID**: Employee ID is required
- Age**: Age is required
- Phone Number**: Phone Number is required
- Salary**: Salary is required
- Email Address**: Email Address is required

Type Check

The screenshot shows a window titled "Data Validation" with a dark blue header and a light blue sidebar. The sidebar contains four buttons: "Save", "Clear", "Help", and "Exit". The main area has a dark blue background with several input fields and their corresponding validation rules and error messages.

Field	Value	Validation Rule	Error Message
Full Name	e	Use only letters, spaces, hyphens (2-50 chars)	
Employee ID	dfer	Format: EMP12345 (EMP + 5 digits)	
Age	290	Age must be 18-65	
Phone Number	344444	Must be 10 digits starting with 0	
Salary	32	Salary must be 15,000-500,000	
Email Address	433		Invalid email format

Length Check

The screenshot shows a 'Data Validation' application window. On the left is a dark sidebar with the text 'DATA VALIDATION' and four buttons: 'Save', 'Clear', 'Help', and 'Exit'. The main area has a teal background and contains several input fields with validation feedback:

- Full Name:** The input field contains the letter 'e'. Below it, a red message reads: 'Use only letters, spaces, hyphens (2-50 chars)'.
- Employee ID:** The input field contains 'EMP123456798'. Below it, a red message reads: 'Format: EMP12345 (EMP + 5 digits)'.
- Age:** The input field contains '290'. Below it, a red message reads: 'Age must be 18-65'.
- Phone Number:** The input field contains '344444'. Below it, a red message reads: 'Must be 10 digits starting with 0'.
- Salary:** The input field contains '32'. Below it, a red message reads: 'Salary must be 15,000-500,000'.
- Email Address:** The input field contains '433'. Below it, a red message reads: 'Invalid email format'.

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
 */
package section1;

import java.awt.Color;
import java.awt.event.KeyEvent;
import javax.swing.JOptionPane;

/**
 *
 * @author bompe
 */
public class Form1 extends javax.swing.JFrame {

    /**
     * Creates new form Form1
     */
    public Form1() {
        initComponents();
        setupRealTimeValidation(); // Add this line
        clearAllErrors(); // Add this line
    }

    /**
     * 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() {

        jPanel1 = new javax.swing.JPanel();
        jPanel2 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
        jButton4 = new javax.swing.JButton();
        jLabel8 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
    }
}
```

```

txtFN = new javax.swing.JTextField();
jLabel2 = new javax.swing.JLabel();
txtID = new javax.swing.JTextField();
jLabel3 = new javax.swing.JLabel();
txtEA = new javax.swing.JTextField();
jLabel4 = new javax.swing.JLabel();
txtAge = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel();
txtPN = new javax.swing.JTextField();
jLabel6 = new javax.swing.JLabel();
txtS = new javax.swing.JTextField();
jLabel7 = new javax.swing.JLabel();
jLabelFN = new javax.swing.JLabel();
jLabelEmail = new javax.swing.JLabel();
jLabelAge = new javax.swing.JLabel();
jLabelID = new javax.swing.JLabel();
jLabelPN = new javax.swing.JLabel();
jLabelS = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
setTitle("Data Validation");

jPanel1.setBackground(new java.awt.Color(0, 102, 102));

jPanel2.setBackground(new java.awt.Color(0, 204, 102));

jButton1.setBackground(new java.awt.Color(0, 204, 102));
jButton1.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jButton1.setForeground(new java.awt.Color(255, 255, 255));
jButton1.setText("Save");
jButton1.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jButton1.setContentAreaFilled(false);
jButton1.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT_CURSOR));
jButton1.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton1ActionPerformed(evt);
    }
});

jButton2.setBackground(new java.awt.Color(0, 204, 102));
jButton2.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jButton2.setForeground(new java.awt.Color(255, 255, 255));
jButton2.setText("Clear");

jButton2.setBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED)
);
jButton2.setContentAreaFilled(false);
jButton2.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton2ActionPerformed(evt);
    }
});

jButton3.setBackground(new java.awt.Color(0, 204, 102));
jButton3.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

```

```

jButton3.setForeground(new java.awt.Color(255, 255, 255));
jButton3.setText("Help");
jButton3.setBorder(javax.swing.BorderFactory.createEtchedBorder());
jButton3.setContentAreaFilled(false);
jButton3.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton3ActionPerformed(evt);
    }
});

jButton4.setBackground(new java.awt.Color(0, 204, 102));
jButton4.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jButton4.setForeground(new java.awt.Color(255, 255, 255));
jButton4.setText("Exit");

jButton4.setBorder(javax.swing.BorderFactory.createEtchedBorder(javax.swing.border.EtchedBorder.RAISED)
);
jButton4.setContentAreaFilled(false);
jButton4.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        jButton4ActionPerformed(evt);
    }
});

jLabel8.setIcon(new javax.swing.ImageIcon("C:\\Users\\bompe\\OneDrive\\Desktop\\JavaSwing-Anthony
Bompetshi\\there2.png")); // NOI18N

javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);
jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel2Layout.createSequentialGroup()
                    .addGap(35, 35, 35)
                    .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                        .addComponent(jButton4, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton3, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton2, javax.swing.GroupLayout.DEFAULT_SIZE, 213, Short.MAX_VALUE)
                        .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                    .addComponent(jLabel8))
                .addGap(49, 49, 49)
                .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            )
            .addGap(18, 18, 18)
        )
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
    );
jPanel2Layout.setVerticalGroup(
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(49, 49, 49)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
        )
    );

```

```

        .addComponent(jButton2, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(27, 27, 27)
        .addComponent(jButton3, javax.swing.GroupLayout.PREFERRED_SIZE, 41,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jButton4, javax.swing.GroupLayout.PREFERRED_SIZE, 42,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, Short.MAX_VALUE))
    );

    jLabel1.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabel1.setForeground(new java.awt.Color(0, 204, 102));
    jLabel1.setText("Full Name");

    txtFN.setBackground(new java.awt.Color(0, 102, 102));
    txtFN.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    txtFN.setForeground(new java.awt.Color(255, 255, 255));
    txtFN.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));
    txtFN.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtFNActionPerformed(evt);
        }
    });

    txtID.setBackground(new java.awt.Color(0, 102, 102));
    txtID.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    txtID.setForeground(new java.awt.Color(255, 255, 255));
    txtID.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));
    txtID.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtIDActionPerformed(evt);
        }
    });

    jLabel3.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabel3.setForeground(new java.awt.Color(0, 204, 102));
    jLabel3.setText("Phone Number");

    txtEA.setBackground(new java.awt.Color(0, 102, 102));
    txtEA.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    txtEA.setForeground(new java.awt.Color(255, 255, 255));
    txtEA.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));
    txtEA.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            txtEAActionPerformed(evt);
        }
    });

    jLabel4.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
    jLabel4.setForeground(new java.awt.Color(0, 204, 102));
    jLabel4.setText("Employee ID");

    txtAge.setBackground(new java.awt.Color(0, 102, 102));
    txtAge.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N

```

```

txtAge.setForeground(new java.awt.Color(255, 255, 255));
txtAge.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));
txtAge.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtAgeActionPerformed(evt);
    }
});

jLabel5.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel5.setForeground(new java.awt.Color(0, 204, 102));
jLabel5.setText("Age");

txtPN.setBackground(new java.awt.Color(0, 102, 102));
txtPN.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
txtPN.setForeground(new java.awt.Color(255, 255, 255));
txtPN.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));

jLabel6.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel6.setForeground(new java.awt.Color(0, 204, 102));
jLabel6.setText("Salary");

txtS.setBackground(new java.awt.Color(0, 102, 102));
txtS.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
txtS.setForeground(new java.awt.Color(255, 255, 255));
txtS.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 204, 102), 3, true));

jLabel7.setFont(new java.awt.Font("Segoe UI", 1, 14)); // NOI18N
jLabel7.setForeground(new java.awt.Color(0, 204, 102));
jLabel7.setText("Email Address");

jLabelFN.setForeground(new java.awt.Color(0, 102, 102));
jLabelFN.setText("jLabel9");

jLabelEmail.setForeground(new java.awt.Color(0, 102, 102));
jLabelEmail.setText("jLabel10");

jLabelAge.setForeground(new java.awt.Color(0, 102, 102));
jLabelAge.setText("jLabel11");

jLabelID.setForeground(new java.awt.Color(0, 102, 102));
jLabelID.setText("jLabel12");

jLabelPN.setForeground(new java.awt.Color(0, 102, 102));
jLabelPN.setText("jLabel13");

jLabelS.setForeground(new java.awt.Color(0, 102, 102));
jLabelS.setText("jLabel14");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(
    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED_SIZE, 297,
                javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

.addGap(51, 51, 51)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addComponent(jLabel2)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
.addComponent(jLabelFN))
.addComponent(jLabel1)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(txtID, javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel4)
.addComponent(jLabel3)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(17, 17, 17)
.addComponent(jLabelID)))
.addGap(24, 24, 24)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabel6)
.addComponent(txtAge, javax.swing.GroupLayout.PREFERRED_SIZE, 184,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(txtS, javax.swing.GroupLayout.PREFERRED_SIZE, 184,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel5)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(6, 6, 6)

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabelAge)
.addComponent(jLabelS))))))
.addGap(24, 56, Short.MAX_VALUE))
.addGroup(jPanel1Layout.createSequentialGroup())
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(txtFN, javax.swing.GroupLayout.PREFERRED_SIZE, 426,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(txtPN, javax.swing.GroupLayout.PREFERRED_SIZE, 218,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(txtEA, javax.swing.GroupLayout.PREFERRED_SIZE, 426,
javax.swing.GroupLayout.PREFERRED_SIZE)
.addComponent(jLabel7)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(6, 6, 6)
.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addComponent(jLabelPN)
.addComponent(jLabelEmail))))
.addGap(0, 0, Short.MAX_VALUE)))
);
jPanel1Layout.setVerticalGroup(
jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(jPanel1Layout.createSequentialGroup())
.addGap(51, 51, 51)
.addComponent(jLabel1)
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```

```

        .addComponent(txtFN, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabelFN)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel2)
        .addGap(43, 43, 43)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel4)
            .addComponent(jLabel5))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtID, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtAge, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabelAge)
            .addComponent(jLabelID))
        .addGap(26, 26, 26)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabel3)
            .addComponent(jLabel6))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(txtPN, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(txtS, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
            .addComponent(jLabelPN)
            .addComponent(jLabelS))
        .addGap(31, 31, 31)
        .addComponent(jLabel7)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(txtEA, javax.swing.GroupLayout.PREFERRED_SIZE, 40,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabelEmail)
        .addContainerGap(66, Short.MAX_VALUE))
        .addComponent(jPanel2, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                .addContainerGap())
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE, Short.MAX_VALUE)
                .addContainerGap())
    );

```



```

        .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );

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

private void txtFNActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtAgeActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtEAActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void txtIDActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    new Form2().setVisible(true);
    dispose();
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtFN.setText("");
    txtID.setText("");
    txtAge.setText("");
    txtPN.setText("");
    txtS.setText("");
    txtEA.setText("");

    // Clear errors using the method
    clearAllErrors();
}

private void clearAllErrors() {
    jLabelFN.setText(".");
    jLabelFN.setForeground(new Color(0, 102, 102));

    jLabelID.setText(".");
    jLabelID.setForeground(new Color(0, 102, 102));

    jLabelAge.setText(".");
    jLabelAge.setForeground(new Color(0, 102, 102));

    jLabelS.setText(".");
    jLabelS.setForeground(new Color(0, 102, 102));
}

```

```

jLabelPN.setText(".");
jLabelPN.setForeground(new Color(0, 102, 102));

jLabelEmail.setText(".");
jLabelEmail.setForeground(new Color(0, 102, 102));

// Also reset text field backgrounds
txtFN.setBackground(new Color(0, 102, 102));
txtID.setBackground(new Color(0, 102, 102));
txtAge.setBackground(new Color(0, 102, 102));
txtS.setBackground(new Color(0, 102, 102));
txtPN.setBackground(new Color(0, 102, 102));
txtEA.setBackground(new Color(0, 102, 102));

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    String NameInput=txtFN.getText();
    String AgeInput=txtAge.getText();
    String IdInput=txtID.getText();
    String SalaryInput=txtS.getText();
    String CellInput=txtPN.getText();
    String EmailInput=txtEA.getText();

    // Reset all error messages first

    boolean hasErrors = false;

    // Check each field individually using isBlank()
    if (NameInput.isBlank()) {
        jLabelFN.setText("Full Name is required");
        jLabelFN.setForeground(Color.RED);
        hasErrors = true;
    }

    if (IdInput.isBlank()) {
        jLabelID.setText("Employee ID is required");
        jLabelID.setForeground(Color.RED);
        hasErrors = true;
    }

    if (AgeInput.isBlank()) {
        jLabelAge.setText("Age is required");
        jLabelAge.setForeground(Color.RED);
        hasErrors = true;
    }

    if (SalaryInput.isBlank()) {
        jLabelS.setText("Salary is required");
        jLabelS.setForeground(Color.RED);
        hasErrors = true;
    }
}

```

```

    }

    if (CellInput.isBlank()) {
        jLabelPN.setText("Phone Number is required");
        jLabelPN.setForeground(Color.RED);
        hasErrors = true;
    }

    if (EmailInput.isBlank()) {
        jLabelEmail.setText("Email Address is required");
        jLabelEmail.setForeground(Color.RED);
        hasErrors = true;
    }

    // If no blank fields, proceed with format validation
    if (!hasErrors) {
        validateFormats();
    }
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    int response = JOptionPane.showConfirmDialog(
        this,
        "Are you sure you want to exit?",
        "Exit Application",
        JOptionPane.YES_NO_OPTION,
        JOptionPane.QUESTION_MESSAGE
    );

    if (response == JOptionPane.YES_OPTION) {
        System.exit(0); // Terminate the application
    }
}

private void setupRealTimeValidation() {
    // Full Name validation
    txtFN.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            validateFullNameRealTime();
            checkFormValidity();
        }
    });

    // Employee ID validation
    txtID.addKeyListener(new java.awt.event.KeyAdapter() {
        public void keyReleased(java.awt.event.KeyEvent evt) {
            validateEmployeeIDRealTime();
            checkFormValidity();
        }
    });

    // Age validation - restrict to numbers only
    txtAge.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

public void keyReleased(java.awt.event.KeyEvent evt) {
    validateAgeRealTime();
    checkFormValidity();
}

public void keyTyped(java.awt.event.KeyEvent e) { // CHANGED: Added full package name
    // Restrict to numbers only
    char c = e.getKeyChar();
    if (!Character.isDigit(c) && c != java.awt.event.KeyEvent.VK_BACK_SPACE && c !=
java.awt.event.KeyEvent.VK_DELETE) {
        e.consume();
    }
}
});

// Salary validation
txtS.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        validateSalaryRealTime();
        checkFormValidity();
    }

    public void keyTyped(java.awt.event.KeyEvent e) { // CHANGED: Added full package name
        // Allow numbers and decimal point
        char c = e.getKeyChar();
        if (!Character.isDigit(c) && c != '.' && c != java.awt.event.KeyEvent.VK_BACK_SPACE && c !=
java.awt.event.KeyEvent.VK_DELETE) {
            e.consume();
        }
        // Allow only one decimal point
        if (c == '.' && txtS.getText().contains(".")) {
            e.consume();
        }
    }
});

// Phone Number validation - restrict to numbers only
txtPN.addKeyListener(new java.awt.event.KeyAdapter() {
    public void keyReleased(java.awt.event.KeyEvent evt) {
        validatePhoneRealTime();
        checkFormValidity();
    }

    public void keyTyped(java.awt.event.KeyEvent e) { // CHANGED: Added full package name
        // Restrict to numbers only
        char c = e.getKeyChar();
        if (!Character.isDigit(c) && c != java.awt.event.KeyEvent.VK_BACK_SPACE && c !=
java.awt.event.KeyEvent.VK_DELETE) {
            e.consume();
        }
    }
});

// Email validation
txtEA.addKeyListener(new java.awt.event.KeyAdapter() {

```

```

public void keyReleased(java.awt.event.KeyEvent evt) {
    validateEmailRealTime();
    checkFormValidity();
}
});
}

private void validateFullNameRealTime() {
    String text = txtFN.getText().trim();

    if (text.isEmpty()) {
        jLabelFN.setText("Full Name is required");
        jLabelFN.setForeground(Color.RED);
    } else if (!text.matches("[a-zA-Z\\s\\-']{2,50}")) {
        jLabelFN.setText("Use only letters, spaces, hyphens (2-50 chars)");
        jLabelFN.setForeground(Color.RED);
    } else {
        jLabelFN.setText("✓ Valid");
        jLabelFN.setForeground(Color.GREEN);
        txtFN.setBackground(new Color(0, 102, 102));
    }
}

private void validateEmployeeIDRealTime() {
    String text = txtID.getText().trim();

    if (text.isEmpty()) {
        jLabelID.setText("Employee ID is required");
        jLabelID.setForeground(Color.RED);
    } else if (!text.matches("EMP\\d{5}")) {
        jLabelID.setText("Format: EMP12345 (EMP + 5 digits)");
        jLabelID.setForeground(Color.RED);
    } else {
        jLabelID.setText("✓ Valid");
        jLabelID.setForeground(Color.GREEN);
        txtID.setBackground(new Color(0, 102, 102));
    }
}

private void validateAgeRealTime() {
    String text = txtAge.getText().trim();

    if (text.isEmpty()) {
        jLabelAge.setText("Age is required");
        jLabelAge.setForeground(Color.RED);
    } else {
        try {
            int age = Integer.parseInt(text);
            if (age < 18 || age > 65) {
                jLabelAge.setText("Age must be 18-65");
            }
        }
    }
}

```

```

        jLabelAge.setForeground(Color.RED);

    } else {
        jLabelAge.setText("✓ Valid");
        jLabelAge.setForeground(Color.GREEN);
        txtAge.setBackground(new Color(0, 102, 102));
    }
} catch (NumberFormatException e) {
    jLabelAge.setText("Age must be a number");
    jLabelAge.setForeground(Color.RED);
}
}
}

private void validateSalaryRealTime() {
    String text = txtS.getText().trim();

    if (text.isEmpty()) {
        jLabelS.setText("Salary is required");
        jLabelS.setForeground(Color.RED);
    } else {
        try {
            double salary = Double.parseDouble(text);
            if (salary < 15000 || salary > 500000) {
                jLabelS.setText("Salary must be 15,000-500,000");
                jLabelS.setForeground(Color.RED);
            } else {
                jLabelS.setText("✓ Valid");
                jLabelS.setForeground(Color.GREEN);
                txtS.setBackground(new Color(0, 102, 102));
            }
        } catch (NumberFormatException e) {
            jLabelS.setText("Salary must be a number");
            jLabelS.setForeground(Color.RED);
        }
    }
}

private void validatePhoneRealTime() {
    String text = txtPN.getText().trim();

    if (text.isEmpty()) {
        jLabelPN.setText("Phone number is required");
        jLabelPN.setForeground(Color.RED);
    } else if (!text.matches("0\\d{9}")) {
        jLabelPN.setText("Must be 10 digits starting with 0");
        jLabelPN.setForeground(Color.RED);
    } else {
        jLabelPN.setText("✓ Valid");
    }
}

```

```

        jLabelPN.setForeground(Color.GREEN);
        txtPN.setBackground(new Color(0, 102, 102));
    }
}

private void validateEmailRealTime() {
    String text = txtEA.getText().trim();

    if (text.isEmpty()) {
        jLabelEmail.setText("Email is required");
        jLabelEmail.setForeground(Color.RED);
    } else if (!text.matches("^[A-Za-z0-9+_.-]+@(.+)$")) {
        jLabelEmail.setText("Invalid email format");
        jLabelEmail.setForeground(Color.RED);
    } else {
        jLabelEmail.setText("✓ Valid");
        jLabelEmail.setForeground(Color.GREEN);
        txtEA.setBackground(new Color(0, 102, 102));
    }
}

private void checkFormValidity() {
    boolean allValid =
        !txtFN.getText().trim().isEmpty() && txtFN.getText().matches("[a-zA-Z\\s\\-']{2,50}") &&
        !txtID.getText().trim().isEmpty() && txtID.getText().matches("EMP\\d{5}") &&
        !txtAge.getText().trim().isEmpty() && isValidAgeValue(txtAge.getText().trim()) &&
        !txtS.getText().trim().isEmpty() && isValidSalaryValue(txtS.getText().trim()) &&
        !txtPN.getText().trim().isEmpty() && txtPN.getText().matches("0\\d{9}") &&
        !txtEA.getText().trim().isEmpty() && txtEA.getText().matches("^[A-Za-z0-9+_.-]+@(.+)$");

    jButton1.setEnabled(allValid);
}

private boolean isValidAgeValue(String text) {
    try {
        int age = Integer.parseInt(text);
        return age >= 18 && age <= 65;
    } catch (NumberFormatException e) {
        return false;
    }
}

private boolean isValidSalaryValue(String text) {
    try {
        double salary = Double.parseDouble(text);
        return salary >= 15000 && salary <= 500000;
    } catch (NumberFormatException e) {
        return false;
    }
}

private void validateFormats() {
    boolean allValid = true;

```

```

// Full Name validation (letters, spaces, hyphens, 2-50 chars)
if (!txtFN.getText().matches("[a-zA-Z\\s\\-']{2,50}")) {
    jLabelFN.setText("Invalid name format (letters/spaces only, 2-50 chars)");
    jLabelFN.setForeground(Color.RED);

    allValid = false;
}

// Employee ID validation (EMP12345 format)
if (!txtID.getText().matches("EMP\\d{5}")) {
    jLabelID.setText("Format must be EMP12345");
    jLabelID.setForeground(Color.RED);

    allValid = false;
}

// Age validation (18-65)
try {
    int age = Integer.parseInt(txtAge.getText());
    if (age < 18 || age > 65) {
        jLabelAge.setText("Age must be 18-65");
        jLabelAge.setForeground(Color.RED);

        allValid = false;
    }
} catch (NumberFormatException e) {
    jLabelAge.setText("Age must be a number");
    jLabelAge.setForeground(Color.RED);

    allValid = false;
}

// Salary validation (15000-500000)
try {
    double salary = Double.parseDouble(txtS.getText());
    if (salary < 15000 || salary > 500000) {
        jLabelS.setText("Salary must be 15,000-500,000");
        jLabelS.setForeground(Color.RED);

        allValid = false;
    }
} catch (NumberFormatException e) {
    jLabelS.setText("Salary must be a number");
    jLabelS.setForeground(Color.RED);

    allValid = false;
}

// Phone validation (10 digits starting with 0)
if (!txtPN.getText().matches("0\\d{9}")) {
    jLabelPN.setText("Phone must be 10 digits starting with 0");
    jLabelPN.setForeground(Color.RED);

    allValid = false;
}

```



```

    }

    // Email validation (basic format)
    if (!txtEA.getText().matches("[A-Za-z0-9+_.-]+@(.+)$")) {
        jLabelEmail.setText("Invalid email format");
        jLabelEmail.setForeground(Color.RED);

        allValid = false;
    }

    // If all validations pass
    if (allValid) {
        JOptionPane.showMessageDialog(this, "Form submitted successfully!", "Success",
JOptionPane.INFORMATION_MESSAGE);
        // Clear form after success
        jButton2ActionPerformed(null);
    }
}

/**
 * @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.getInstalledLookAndFeels()) {
            if ("Nimbus".equals(info.getName())) {
                javax.swing.UIManager.setLookAndFeel(info.getClassName());
                break;
            }
        }
    } catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(Form1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(Form1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(Form1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(Form1.class.getName()).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 Form1().setVisible(true);
    }
}

```

```

    });
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JButton jButton2;
private javax.swing.JButton jButton3;
private javax.swing.JButton jButton4;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabelAge;
private javax.swing.JLabel jLabelEmail;
private javax.swing.JLabel jLabelFN;
private javax.swing.JLabel jLabelID;
private javax.swing.JLabel jLabelPN;
private javax.swing.JLabel jLabelS;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JTextField txtAge;
private javax.swing.JTextField txtEA;
private javax.swing.JTextField txtFN;
private javax.swing.JTextField txtID;
private javax.swing.JTextField txtPN;
private javax.swing.JTextField txtS;
// End of variables declaration
}

```