

Form Validation

Email

```
public boolean chekEmail(String email) {  
    int atpos = email.indexOf("@");  
    int dotpos = email.lastIndexOf(".");  
    if (atpos > 1 && (dotpos - atpos) > 2 && dotpos < email.length()) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

Add Button

```
//id  
if (txtId.getText().trim().length() < 5) {  
    JOptionPane.showMessageDialog(null, "Enter your Id");  
  
    //name  
} else if (txtINa.getText().trim().length() < 4) {  
    JOptionPane.showMessageDialog(null, "Enter your Name");  
  
    //age
```

```

    } else if (Integer.parseInt(txtAg.getText()) < 18 ||
Integer.parseInt(txtAg.getText()) > 100) {

        JOptionPane.showMessageDialog(null, "Enter your Age");

//email
    } else if (!chekEmail(txtEm.getText()) || txtEm.getText().trim().length() < 1) {
        JOptionPane.showMessageDialog(null, "Enter your Email");

//gender
    } else if (!rMa.isSelected() && !rFe.isSelected()) {
        JOptionPane.showMessageDialog(null, "Select your Gender");

//course
    } else if (!cPHP.isSelected() && !cJ2EE.isSelected() && !cWPA.isSelected()) {
        JOptionPane.showMessageDialog(null, "Select your Course");

//round
    } else if
(cmbR.getItemAt(cmbR.getSelectedIndex()).trim().equalsIgnoreCase("Select A
Round")) {

        JOptionPane.showMessageDialog(null, "Select your Round");

//comments
    } else if (taMsg.getText().trim().length() < 10) {

        JOptionPane.showMessageDialog(null, "At leaset ten Characters");

```

```
} else {  
  
    int id = Integer.parseInt(txtId.getText());  
    String name = txtINa.getText();  
    int age = Integer.parseInt(txtAg.getText());  
    String email = txtEm.getText();  
  
    String gender = " ";  
    if (rMa.isSelected()) {  
        gender = rMa.getText();  
    }  
    if (rFe.isSelected()) {  
        gender = rFe.getText();  
    }  
  
    String course = " ";  
    if (cPHP.isSelected()) {  
        course += " " + cPHP.getText();  
    }  
    if (cJ2EE.isSelected()) {  
        course += " " + cJ2EE.getText();  
    }  
    if (cWPA.isSelected()) {
```

```
course += " " + cWPA.getText();  
}
```

```
String round = cmbR.getItemAt(cmbR.getSelectedIndex());  
String comments = taMsg.getText();
```

```
String rs = id + "\n" + name + "\n" + age + "\n" + email + "\n" + gender +  
"\n" + course + "\n" + round + "\n" + comments;  
System.out.println(rs);
```

```
DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();  
Object[] row = new Object[8];  
row[0] = id;  
row[1] = name;  
row[2] = age;  
row[3] = email;  
row[4] = gender;  
row[5] = course;  
row[6] = round;  
row[7] = comments;  
model.addRow(row);
```

```
Student s = new Student(id, name, age, email, gender, course, round,  
comments);  
List<Student> list = new ArrayList<>();
```

```
list.add(s);  
try {  
    Utils.writeToFile("motin", list);  
} catch (Exception e) {  
}  
lblMsg.setText("Succesfully added into table and write file");  
}  
}
```

Clear Form

```
txtId.setText("0");  
txtINa.setText("");  
txtAg.setText("0");  
txtEm.setText("");  
buttonGroup1.clearSelection();  
cPHP.setSelected(false);  
cJ2EE.setSelected(false);  
cWPA.setSelected(false);  
cmbR.setSelectedIndex(0);  
taMsg.setText("");
```

Clear Table

```
DefaultTableModel model = (DefaultTableModel) tblDisplay.getModel();  
    model.setRowCount(0);
```

Read From File

```
DefaultTableModel tableModel = (DefaultTableModel) tblDisplay.getModel();  
    tableModel.setRowCount(0);  
    Utils.displayStudentDataFromFile("motin", tableModel);
```

Utils Class

```
public class Utils {  
    public static void writeToFile (String filename, List<Student> student){  
        File destFile = new File(filename+".txt");  
        try {  
            if(destFile.exists()==false){  
                System.out.println("make a new file");  
                destFile.createNewFile();  
            }  
  
            PrintWriter out = new PrintWriter(new FileWriter(destFile,true));  
            for(Student s: student){  
                out.append(s.getId()+"", "+s.getName()+"", "+s.getAge()+"",  
"+s.getEmail()+"", "+s.getGender()+"", "+s.getCourse()+"", "+
```

```

        s.getRound()+"", "+s.getComments()+"\n");
    }
    out.close();
} catch (Exception e) {
}
}
}

```

```

public static void displayStudentDataFromFile(String filename,
DefaultTableModel model){
    String line;
    try {
        BufferedReader reader = new BufferedReader(new
        FileReader(filename+".txt"));
        while((line= reader.readLine()) != null){
            model.addRow(line.split(" "));
        }
        reader.close();
    } catch (Exception e) {
    }
}
}
}

```