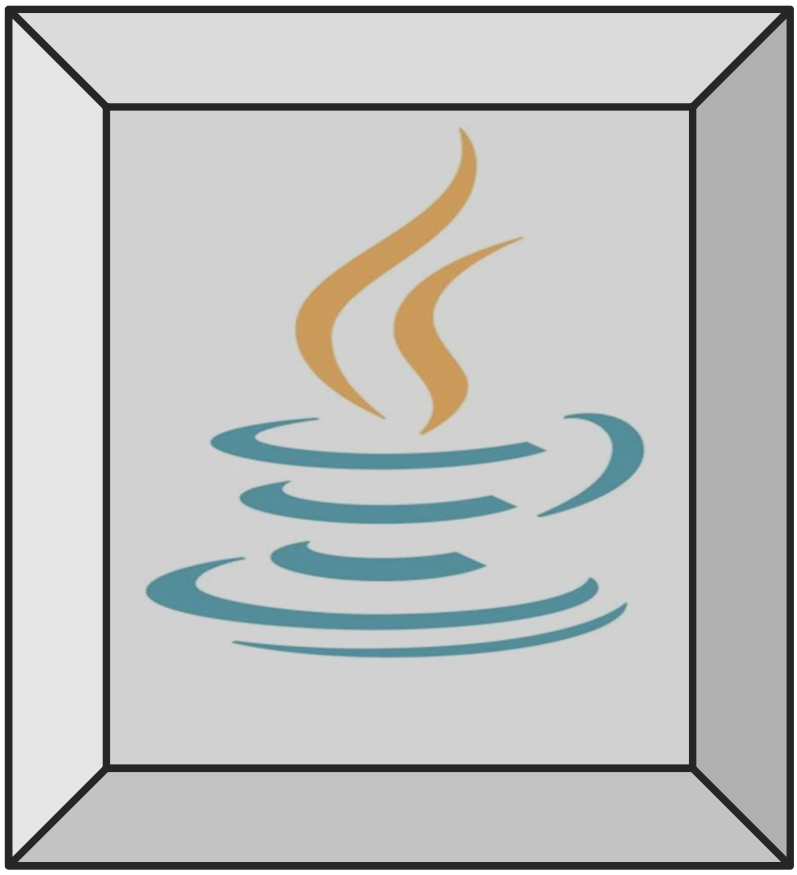
JAVA PROJECT

FILE



NAME: CHETNA & MADHU

ROLL NO:

SUBMITTED TO: MRS. SUMAN VERMA

ACKNOWELDGMENT

I would like to express my special thanks of gratitude to my Information Technology teacher Mrs. Suman Verma as well as our principal Mrs. Manjeet Gill who gave me the golden opportunity to do this wonderful project on the topic Java.

Secondly I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

Lastly, I like to thank all my supporters hi have motivated me to fulfil their project before the timeline.

CONTENT

INTRODUCTION TO JAVA HISTORY

The history of Java starts with the Green Team. Java team members (also known as Green Team), initiated this project to develop a language for digital devices such as set-top boxes, televisions, etc. However, it was best suited for internet programming. Later, Java technology was incorporated by Netscape. Java was developed by James Gosling, who is known as the father of Java, in 1995. James Gosling and his team members started the project in the early '90s.

Over the years, the language has been the foundation of millions of applications across many platforms, such as Windows, Macintosh, UNIX, Android-based handheld devices, Embedded Systems, and corporate solutions.

JAVA AND OPEN SOURCE

Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System. On November 13, 2006, Sun released much of its Java virtual machine (JVM) as free and open-source software (FOSS), under the terms of the GPL-2.0-only license.

JAVA VIRTUAL MACHINE

A **Java virtual machine** (**JVM**) is a virtual machine that enables a computer to run Java programs as well as programs written in other languages that are also compiled to Java Bytecode. The JVM is detailed by a specification that formally describes what is required in a JVM implementation. Having a specification ensures interoperability of Java programs across different implementations so that program authors using the Java Development Kit (JDK) need not worry about idiosyncrasies of the underlying hardware platform.

The JVM reference implementation is developed by the OpenJDK project as open source code. The commercially supported Java releases available from Oracle are based on the OpenJDK runtime. Eclipse OpenJ9 is another open source JVM for OpenJDK.

CHARACTERISTICS OF JAVA

- Java is simple
- Object-Oriented
- Platform Independent
- Secure
- Robust
- Architectural Neutral
- Portable
- Performance
- Distributed
- Multithreaded
- Java is dynamic

INTRODUCTION TO NETBEANS IDE

NetBeans IDE is used to create java applications very easily using the efficient GUI builder. It allows us to develop applications by dragging and positioning GUI components from a palette onto a container. The GUI builder automatically takes care of the correct spacing and alignment of the different components relative to each other.

- **GUI builder :** It is an area to place components on the form visually. There are two views of the GUI builder
    1. Design View
    2. Source View
- **Palette :** Palette contains controls or components use to create GUI applications.

- **Code Editor Window :** It is an area where we write code for our java application.

- **Inspector Window :** This window is used to display a hierarchy of all the components or controls placed on the current form.

- **Property Window :** Using this window we can make changes in the property of currently selected controls on the form.

COMPONENTS

Components are also known as ("widgets") are the basic interface elements the user interacts with various components such as –

- jButton
- jlabel
- jTextField
- jTextArea
- jPassword
- jComboBox
- jCheckBox
- jOptionPane
- jList

Components are placed on a container (like the jFrame)

There are two types of controls:

1. **Parent or Container Control-** They act as a background for other controls. For ex- Frame. When we delete or move along with it.
2. **Child Control-** Controls placed inside a container control are called Child Control. For ex-TextField.

DEFINITION OF COMPONENTS

- **jFrame :-** The jFrame is a window with title, border, (optional) menu bar and is used to contain all other components placed by the user on the form.

  - **Property :-**
    1. defaultCloseOperation
    2. Title

- **jButton :-** A button is a component that the user presses or pushes to trigger a specific action. When the user clicks on the button at runtime, the code associated with the click action gets executed.

  - **Property :-**
    1. Background
    2. Enabled
    3. Font
    4. Foreground
    5. Horizontal alignment

  - **Methods :-**
    1. getText()
    2. setText()

- **jlabel :-** It provides text instructions or information and it also display a single line read-only text, an image or both text and image.

  - **Property :-**
    1. Background
    2. Enabled
    3. Font
    4. Editable
    5. toolTipText

  - **Methods :-**
    1. isEditable()
    2. isEnabled()

- **jTextField :-** It allows editing/displaying of a single line of text. It is an input area where the user a type in characters or numbers.

  - **Property :-**
    1. Background
    2. Enabled
    3. Font
    4. Foreground

➲ **Methods :-**

1. getText()
2. setText()
3. isEnabled()

TextArea :- This component allow us to accept multiline input from the user or display multiple lines of information. This component automatically adds vertical or horizontal scrollbars and when during run time.

➲ **Property :-**

1. Background
2. lineWrap
3. Rows
4. wrapStyleWord
5. Columns

➲ **Methods :-**

1. append ()
2. getText()
3. setText()
4. isEditable()
5. isEnabled()

- **jPassword :-** It is used to enter confidential input like passwords. Each character entered can be replaced by an echo character. By default, the echo character is the asterisk (*).

  - **Property :-**
    1. Background
    2. echoChar
    3. Font
    4. Foreground
    5. Text

- **jRadioButton :-** It is used to provide the user several choices and allow him/her to select one of the choices.

  - **Property :-**
    1. Background
    2. Selected
    3. Font
    4. Foreground
    5. buttonGroup

  - **Methods :-**
    1. getText()
    2. setText()
    3. isSelected()
    4. setSelected()

- **jCheckBox :-** It is a small box like component that is either marked or unmarked. When it is clicked, it changes from checked to unchecked or vice versa automatically.

  - **Property :-**
    1. Background
    2. Selected
    3. Foreground
    4. buttonGroup

  - **Methods :-**
    1. getText()
    2. setText()
    3. isSelected()
    4. setSelected()

- **jComboBox:-** It is like a drop down box-you can click a drop-down arrow and select an option from a list.

  - **Property :-**
    1. background
    2. selectIndex
    3. model
    4. buttonGroup
    5. selectionMode

➲ **Methods :-**

1. getSelectedValue()
2. isSelectedIndex()

jList :- It provides a scrollable set of items from which one or more may be selected.

➲ **Property :-**

1. background
2. selectIndex
3. model
4. buttonGroup
5. Selectitem

➲ **Methods :-**

1. getSelectedItem()
2. isSelectedIndex()
3. setModel()

JOptionPane :- we use this component when we want to request information from the user. It requires an import statement at the top of the program-

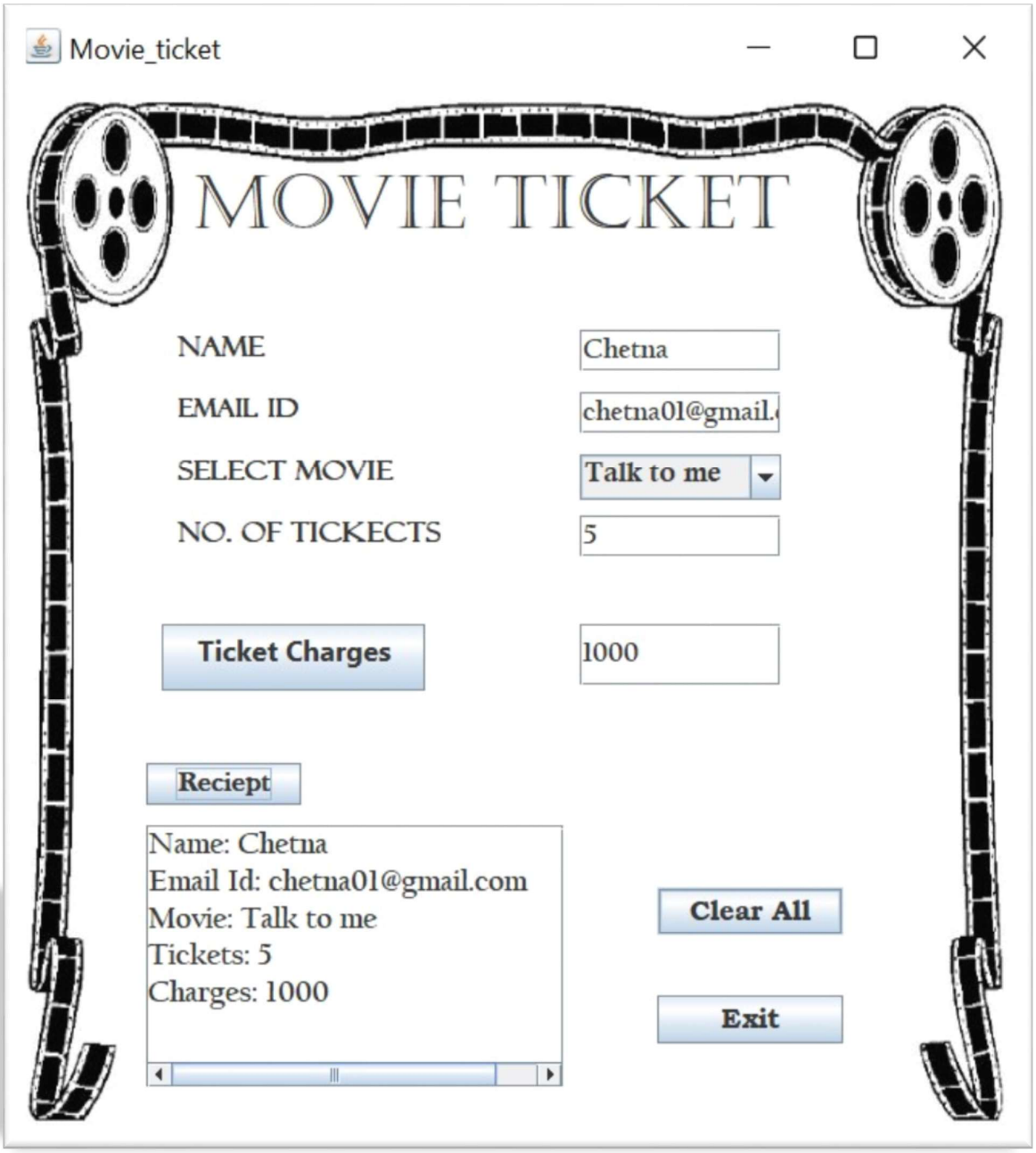
```
Import java.swing.JOptionPane;
```

➲ **Methods :-**

1. showMessageDialog()
2. showConfirmDialog()
3. showInputDialog()

Programs With GUI

1. Movie Ticket

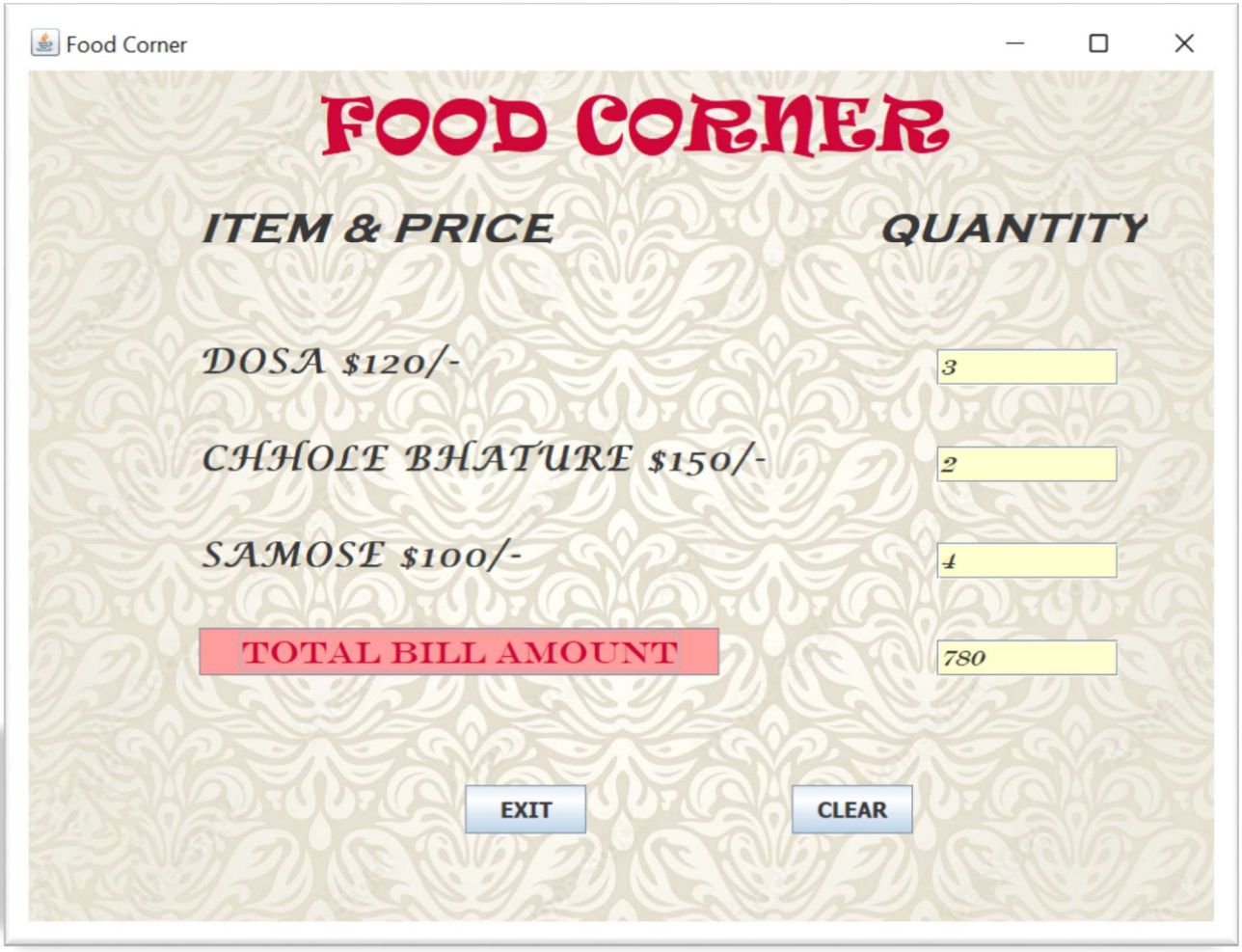


Source Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {


int mov , tix;
mov = jComboBox1.getSelectedIndex();
tix = Integer.parseInt(jTextField3.getText());
jTextField4.setText(String.valueOf(mov=200*tix));
}


private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
String name = jTextField1.getText();
String id = jTextField2.getText();
String mov = (String)jComboBox1.getSelectedItem();
int tix = Integer.parseInt(jTextField3.getText());
int charge = Integer.parseInt(jTextField4.getText());
String reciept= &quot;Name: &quot; + name + &quot;\n&quot; +
&quot;Email Id: &quot; + id + &quot;\n&quot; +
&quot;Movie: &quot; + mov + &quot;\n&quot; +
&quot;Tickets: &quot; + tix + &quot;\n&quot; +
&quot;Charges: &quot; + charge;
```

```
jTextArea1.setText(reciept);
}

private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
jTextField1.setText(&quot;&quot;);
jTextField2.setText(&quot;&quot;);
jTextField3.setText(&quot;&quot;);

jTextField4.setText(&quot;&quot;);
jComboBox1.setSelectedIndex(0);
jTextArea1.setText(&quot;&quot;);
}

private void
jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}
```

2. Food Corner



Source Code:

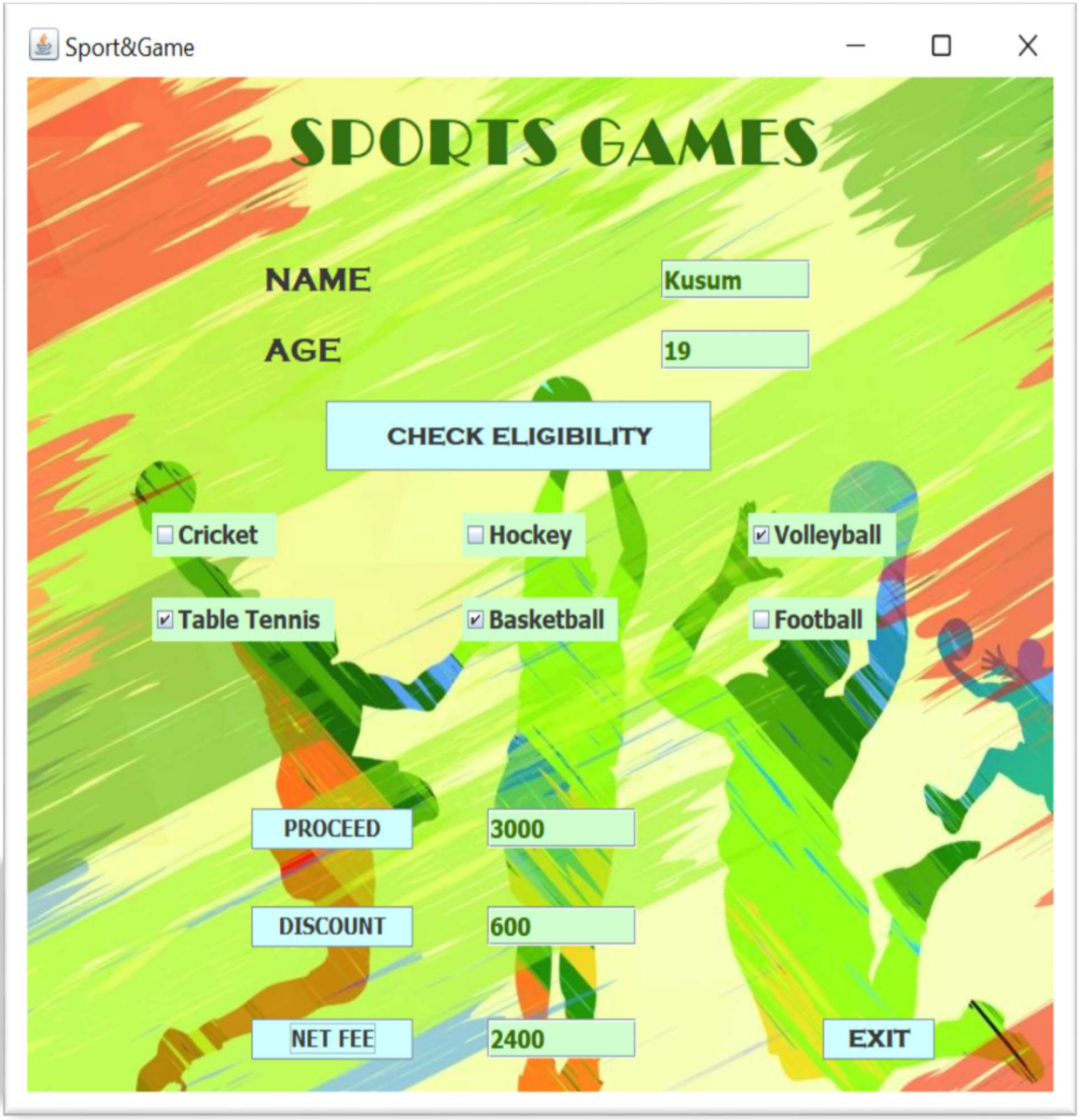
```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

int a, b, c, d;

a = Integer.parseInt(jTextField1.getText());

b = Integer.parseInt(jTextField2.getText());


c = Integer.parseInt(jTextField3.getText());

d = a*120 + b*150 + c*30;

jTextField4.setText(String.valueOf(d));

}


private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);

}


private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {

jTextField1.setText(&amp;quot;&amp;quot;);

jTextField2.setText(&amp;quot;&amp;quot;);

jTextField3.setText(&amp;quot;&amp;quot;);

jTextField4.setText(&amp;quot;&amp;quot;);}
```

3. Blood Eligibilty

Source Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

int age, weight;


age = Integer.parseInt(jTextField1.getText());

weight = Integer.parseInt(jTextField2.getText());

if (age &gt;= 18 &amp;&amp; age &lt; 65 &amp;&amp; weight &gt;=
50)

JOptionPane.showMessageDialog(rootPane,&quot;you are
Eligibile to Donate

Blood&quot;);

else

JOptionPane.showMessageDialog(rootPane,&quot;you are not
Eligibile to Donate

Blood&quot;);

}


private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

System.exit(0);}
```

4. Sports Games

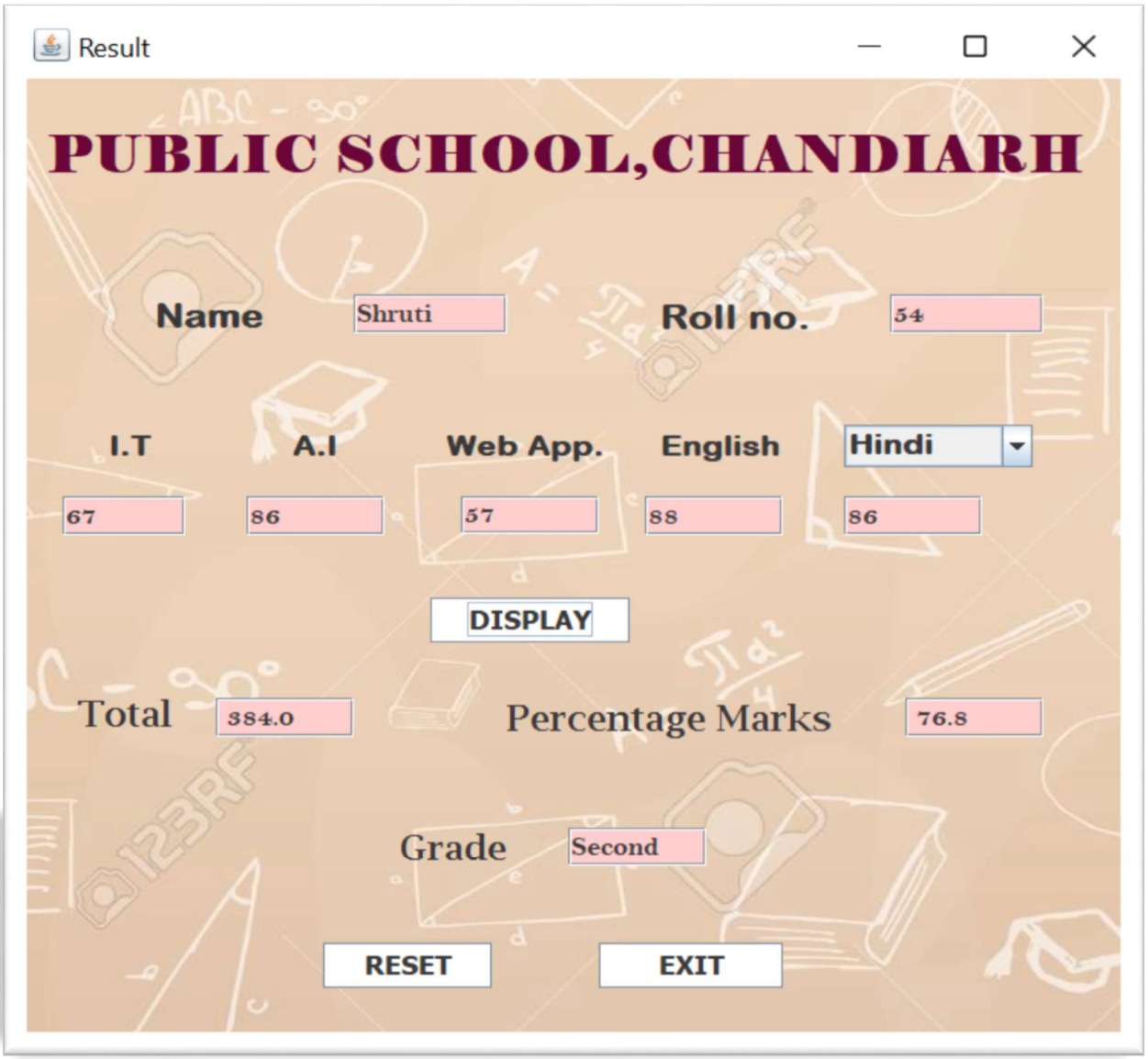


Source Code:

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {

int age=Integer.parseInt(jTextField2.getText());

if (age&gt;=10 &amp;&amp; age&lt;=40) {


JOptionPane.showMessageDialog(rootPane,&quot;Welcome&quo
t;); }

else {

JOptionPane.showMessageDialog(rootPane,&quot;Sorry! You are
either

underage or overage&quot;);

}

}

private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

int game=0;

if(jCheckBox1.isSelected())

game++;

if(jCheckBox2.isSelected())

game++;

if(jCheckBox3.isSelected())

game++;
```

```
if(jCheckBox4.isSelected())
game++;
if(jCheckBox5.isSelected())
game++;
if(jCheckBox6.isSelected())
game++;
jTextField3.setText(&amp;quot;&amp;quot;+(game*1000));
}

private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
int fee=Integer.parseInt(jTextField3.getText());
int disc = 0;
if(fee&amp;gt;1000)

disc=fee*20/100;
jTextField4.setText(&amp;quot;&amp;quot;+(disc));
}

private void
jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
int fee=Integer.parseInt(jTextField3.getText());
int disc=Integer.parseInt(jTextField4.getText());
jTextField5.setText(&amp;quot;&amp;quot;+(fee-disc));}
```

5. Result


Result
PUBLIC SCHOOL,CHANDIARH
Name
Shruti
Roll no.
54
I.T
A.I
Web App.
English
Hindi
67
86
57
88
86
DISPLAY
Total
384.0
Percentage Marks
76.8
Grade
Second
RESET
EXIT

Source Code:

```
private void
jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
double a = Double.parseDouble(jTextField3.getText());
double b = Double.parseDouble(jTextField4.getText());

double c = Double.parseDouble(jTextField5.getText());
double d = Double.parseDouble(jTextField6.getText());
double e = Double.parseDouble(jTextField7.getText());
double x = a + b + c + d + e;
jTextField8.setText(&amp;quot; &amp;quot; + x);
double y = ((a + b + c + d + e) / 500) * 100;
jTextField9.setText(&amp;quot; &amp;quot; + y);
if (y &amp;gt; 90) {
jTextField10.setText(&amp;quot;Distinction&amp;quot;);
} else if (y &amp;gt; 85) {
jTextField10.setText(&amp;quot;Honours&amp;quot;);
} else if (y &amp;gt; 80) {
jTextField10.setText(&amp;quot;First&amp;quot;);
} else if (y &amp;gt; 70) {
jTextField10.setText(&amp;quot;Second&amp;quot;);
} else if (y &amp;gt; 60) {
```

```
jTextField10.setText(&quot;Third&quot;);
} else {
jTextField10.setText(&quot;Poor&quot;);
}
}

private void jButton1ActionPerformed(java.awt.event.ActionEvent
evt) {
jTextField1.setText(&quot;&quot;);
jTextField2.setText(&quot;&quot;);
jTextField3.setText(&quot;&quot;);

jTextField4.setText(&quot;&quot;);
jTextField6.setText(&quot;&quot;);
jTextField7.setText(&quot;&quot;);
jTextField8.setText(&quot;&quot;);
jTextField9.setText(&quot;&quot;);
jTextField10.setText(&quot;&quot;);
}
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
System.exit(0);
}
```