

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELAGAVI 590018**



Project Report on  
**“ATTENDANCE TRACKER”**

By

Navaneet S (1BM24CS183)

Naman M (1BM24CS180)

Mohammed Farhaan (1BM23CS169)

Under the Guidance of  
MONISHA H M  
Professor, Department of CSE  
BMS College of Engineering  
Work carried out at



Department of Computer Science and Engineering  
BMS College of Engineering  
(Autonomous college under VTU)  
P.O. Box No.: 1908, Bull Temple Road, Bangalore-560 019  
2025-2026

**BMS COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



***CERTIFICATE***

This is to certify that the OOPS with JAVA project titled "**Attendance Tracker**" has been carried out by Navaneet S (1BM24CS183), Naman M (1BM24CS180), Mohammed Farhaan(1BM24CS169) during the academic year 2025- 2026.

Signature of the guide

**Monisha H M**

Assistant Professor,

Department of Computer Science and Engineering

BMS College of Engineering, Bangalore

**BMS COLLEGE OF ENGINEERING**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



**DECLARATION**

We, Navaneet S (1BM24CS183), Naman M (1BM24CS180), Mohammed Farhaan (1BM24CS169), students of 3<sup>rd</sup> Semester, B.E, Department of Computer Science and Engineering, BMS College of Engineering, Bangalore, hereby declare that, this project work entitled "ATTENDANCE TRACKER" has been carried out by us under the guidance of Monisha H M, Assistant Professor, Department of CSE, BMS College of Engineering, Bangalore during the academic semester Sep-Dec 2025. We also declare that to the best of our knowledge and belief, the project reported here is not from part of any other report by any other students.

**Signature of the Candidates**

Navaneet S

(1BM24CS183)

Naman M

(1BM24CS180)

Mohammed Farhaan

(1BM24CS169)

## TABLE OF CONTENT

<b>SL NO.</b>	<b>CONTENT TOPIC</b>	<b>PAGE NO.</b>
1	DESCRIPTION OF THE PROJECT	5
2	OBJECTIVE OF THE PROJECT	6
3	INTRODUCTION	7
4	DESIGN MODULE	8
5	DESCRIPTION OF MODULES	9
6	REQUIREMENTS AND FUNCTIONALITY	14
7	JAVA TECHNIQUES USED	15
8	RESULT	16
10	NEW LEARNINGS FROM THE PROJECT	48
11	FUTURE ENHANCEMENTS	49

## **1. DESCRIPTION OF THE PROJECT**

The *Attendance Tracker* is a Java-based mini project developed to simplify and automate the process of maintaining student attendance records. Traditionally, attendance is recorded manually, which is time-consuming, error-prone, and difficult to manage for long periods. This project provides a simple, efficient, and user-friendly solution for managing attendance digitally.

The system allows the user (teacher/admin) to add student details, mark attendance, and store attendance records securely using CSV files. It avoids the need for complex databases and instead uses file handling, making it lightweight and easy to deploy.

The project is developed using Core Java concepts, NetBeans IDE, and file handling techniques. It is suitable for small institutions, labs, or classroom environments where a simple attendance solution is required.

This project demonstrates practical application of Java programming concepts such as classes, objects, file I/O, exception handling, and modular design.

## **2. OBJECTIVE OF THE PROJECT**

The main objectives of this project are:

- To automate the student attendance process
- To reduce manual errors in attendance recording
- To store student and attendance data in digital format
- To provide easy access to attendance records
- To apply Java programming concepts in a real-world application
- To develop a mini project without using a database

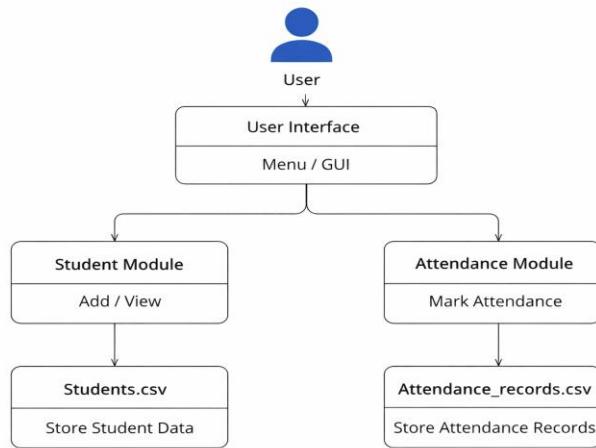
### **3. INTRODUCTION**

Attendance management is an important task in educational institutions. Maintaining attendance records manually can lead to data loss, duplication, and incorrect entries. With the advancement of technology, computer-based attendance systems have become necessary. The Smart Attendance Management System is designed using Java to help teachers manage attendance efficiently. The system stores student details and attendance information in CSV files, ensuring data persistence. It uses simple logic and file handling mechanisms, making it easy to understand and implement. This project is ideal for beginners to understand how Java can be used to build real-time applications using basic concepts.

### **4. DESIGN MODULES**

The project is divided into the following modules:

1. Student Management Module
2. Attendance Management Module
3. File Handling Module
4. User Interface Module
5. Data Storage Module



## 5. DETAILED DESCRIPTION OF MODULES

### 5.1 Student Management Module

This module is responsible for handling student-related operations such as:

- Adding new student details
- Viewing existing student records
- Storing student data in students.csv

It ensures that student information is properly maintained and accessible for attendance marking.

The screenshot shows the Apache NetBeans IDE 28 interface. The left sidebar displays the project structure for "SmartAttendanceSystem [master]". The "nbproject" folder contains "private", "build-impl.xml", "genfiles.properties", "project.properties", and "project.xml". The "src" folder contains "smartattendance" (which has "gui", "logic", and "main.java"), "test", "attendance\_records.csv", "build.xml", "manifest.mf", and "students.csv". The main workspace shows the "Source" tab of the "students.csv" file, which contains the following data:

```
106, Namaj  
107, Farhaan  
108, Sudheep
```

## 5.2 Attendance Management Module

This module allows the user to:

- Mark attendance for students
- Record present/absent status
- Store attendance data in attendance\_records.csv

Attendance records are saved with proper formatting to allow future viewing and analysis.

The screenshot shows the Apache NetBeans IDE interface with the title "SmartAttendanceSystem - Apache NetBeans IDE 28". The left pane displays the project structure under "SmartAttendanceSystem [master]". The "Source" tab in the center pane shows the content of the "attendance\_records.csv" file, which contains the following data:

Index	Date	ID	Status
1	2025-12-30	101	Rahul, Present
2	2025-12-30	102	Aditi, Present
3	2025-12-30	103	Kiran, Present
4	2025-12-30	104	Sneha, Absent
5	2025-12-30	105	Arjun, Absent
6	2025-12-30	105	Arjun, Absent
7	2025-12-30	101	Rahul, Present
8	2025-12-30	102	Aditi, Present
9	2025-12-30	103	Kiran, Present
10	2025-12-30	104	Sneha, Absent
11	2025-12-30	105	Arjun, Absent
12	2025-12-30	101	Naman, Present
13	2025-12-30	102	Megha, Present
14	2025-12-30	103	Monisha, Absent
15	2025-12-30	104	Faraaan, Absent
16	2025-12-30	105	Navaneet, Present
17	2025-12-30	101	Naman, Present
18	2025-12-30	102	Megha, Present
19	2025-12-30	103	Monisha, Absent
20	2025-12-30	104	Faraaan, Absent
21	2025-12-30	105	Navaneet, Present
22			

## 5.3 File Handling Module

This module uses Java File I/O concepts to:

- Read data from CSV files
- Write updated data into CSV files
- Ensure data persistence even after the program exits

It eliminates the need for databases and keeps the project lightweight.

## 5.4 User Interface Module

The user interface is developed using:

- Console-based interaction **or**
- Java Swing components (if applicable)

It allows users to interact with the system easily using menu options and buttons.

## **5.5 Data Storage Module**

This module manages:

- Storage of student records
- Storage of attendance records

CSV files are used as the data source, making the system portable and easy to manage.

## **6. REQUIREMENTS AND FUNCTIONALITY**

### **Software Requirements**

- Java JDK
- NetBeans IDE
- Windows / Linux / macOS

### **Hardware Requirements**

- Minimum 4GB RAM
- Basic system with Java support

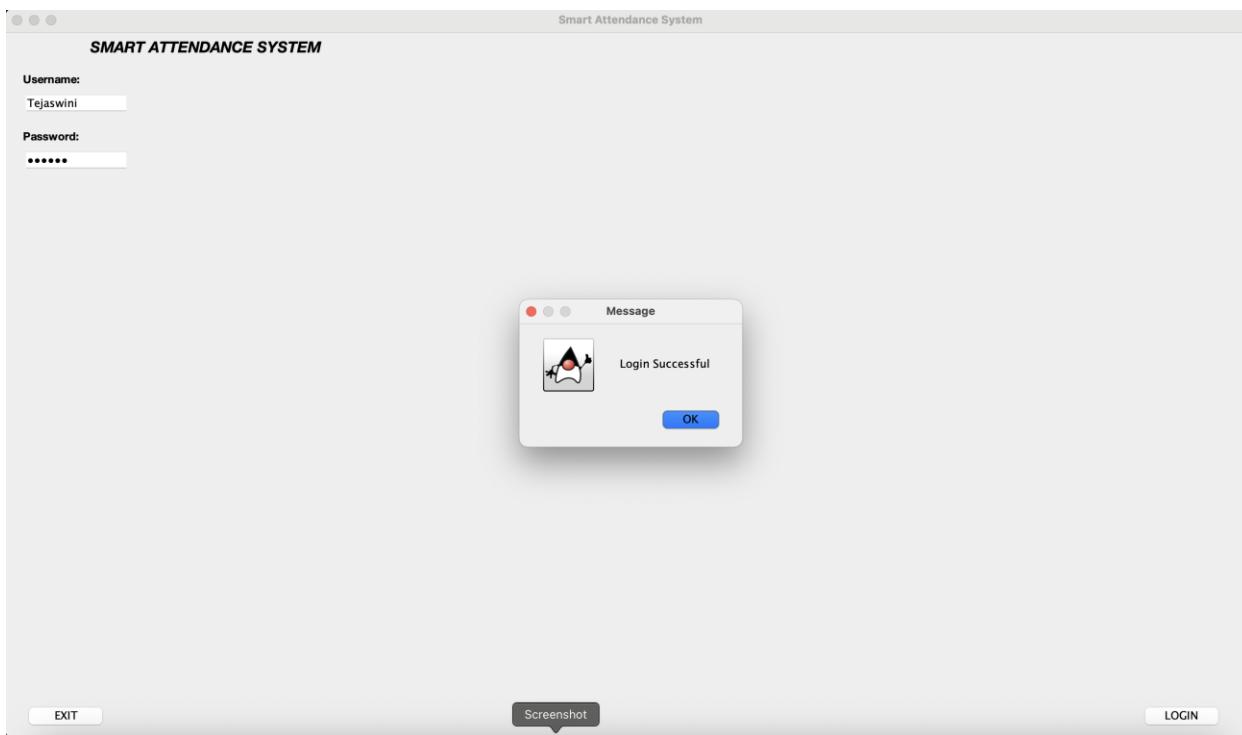
## **7. JAVA TECHNIQUES / MECHANISMS USED**

The following Java concepts are used in the project:

- **Classes and Objects** – To represent students and attendance records
- **Packages** – For better organization of source code

- **File Handling** – Reading and writing CSV files
- **Exception Handling** – Handling runtime errors safely
- **Collections (if used)** – Managing data efficiently
- **Control Statements** – Implementing project logic
- **Swing / Console I/O** – For user interaction

## 8. SCREENSHOTS (RESULT)



This screenshot shows the login interface of the Smart Attendance System. The user enters a valid username and password, and upon successful verification, the system displays a confirmation message. This ensures secure access to the application and restricts unauthorized usage.

### **CODE:**

```
package smartattendance.gui;

import javax.swing.JFrame; import javax.swing.JOptionPane;

/** *

• @author navaneet.s

*/
public class LoginFrame extends javax.swing.JFrame {

private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(LoginFrame.class.getName());

/**
 * Creates new form LoginFrame
 */
public LoginFrame() {
    initComponents();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
}

/**
 * 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()      {

    jLabel1      =      new      javax.swing.JLabel();
    jLabel2      =      new      javax.swing.JLabel();
    txtUsername      =      new      javax.swing.JTextField();
    jLabel3      =      new      javax.swing.JLabel();
    txtPassword      =      new      javax.swing.JPasswordField();
    btnLogin      =      new      javax.swing.JButton();
    btnExit      =      new      javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
    setTitle("Smart      Attendance      System\n");
    setResizable(false);

    jLabel1.setFont(new      java.awt.Font("Helvetica      Neue",      3,      18));      //      NOI18N
    jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
    jLabel1.setText("SMART      ATTENDANCE      SYSTEM      ");

    jLabel2.setFont(new      java.awt.Font("Helvetica      Neue",      1,      13));      //      NOI18N
    jLabel2.setText("Username:");

    txtUsername.setText("NAME");
    txtUsername.addActionListener(this::txtUsernameActionPerformed);

    jLabel3.setFont(new      java.awt.Font("Helvetica      Neue",      1,      13));      //      NOI18N
    jLabel3.setText("Password:");

    txtPassword.setText("password");
    txtPassword.addActionListener(this::txtPasswordActionPerformed);

    btnLogin.setText("LOGIN");
    btnLogin.addActionListener(this::btnLoginActionPerformed);
```

```

btnExit.setText("EXIT");
btnExit.addActionListener(this::btnExitActionPerformed);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(100, 100, 100)
                .addComponent(jLabel1))
            .addGroup(layout.createSequentialGroup()
                .addGap(21, 21, 21)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(100, 100, 100)
                        .addComponent(txtPassword, javax.swing.GroupLayout.DEFAULT_SIZE, 125,
Short.MAX_VALUE))
                    .addComponent(txtUsername)
                    .addComponent(btnExit, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGapContainerGap(100, Short.MAX_VALUE))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup()
    .addGap(38, 38, 38)
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(0, 0, Short.MAX_VALUE)
            .addComponent(btnLogin, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
            .addGap(38, 38, 38)
            .addComponent(btnExit, javax.swing.GroupLayout.PREFERRED_SIZE, 100,
javax.swing.GroupLayout.PREFERRED_SIZE)))
    .addGap(38, 38, 38));
);
layout.setVerticalGroup(

```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel1)
            .addGap(18, 18, 18)
            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 17,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtUsername, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jLabel3)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addComponent(txtPassword, javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 109,
                Short.MAX_VALUE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
            false)
            .addComponent(btnLogin, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE)
            .addComponent(btnExit, javax.swing.GroupLayout.DEFAULT_SIZE,
                Short.MAX_VALUE))
            .addGap(11, 11, 11))
    );
}

```

```

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

```

```

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

```

```

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

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

String username = txtUsername.getText(); String password = new
String(txtPassword.getPassword());

if (username.equals("Tejaswini") && password.equals("bms123")) {
JOptionPane.showMessageDialog(this, "Login Successful");

DashboardFrame db = new DashboardFrame();
db.setVisible(true);
this.dispose(); // close login window
}

} else { JOptionPane.showMessageDialog(this, "Invalid Username or Password"); }

// TODO add your handling code here:
}

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

System.exit(0); // 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
*/
}

```

```

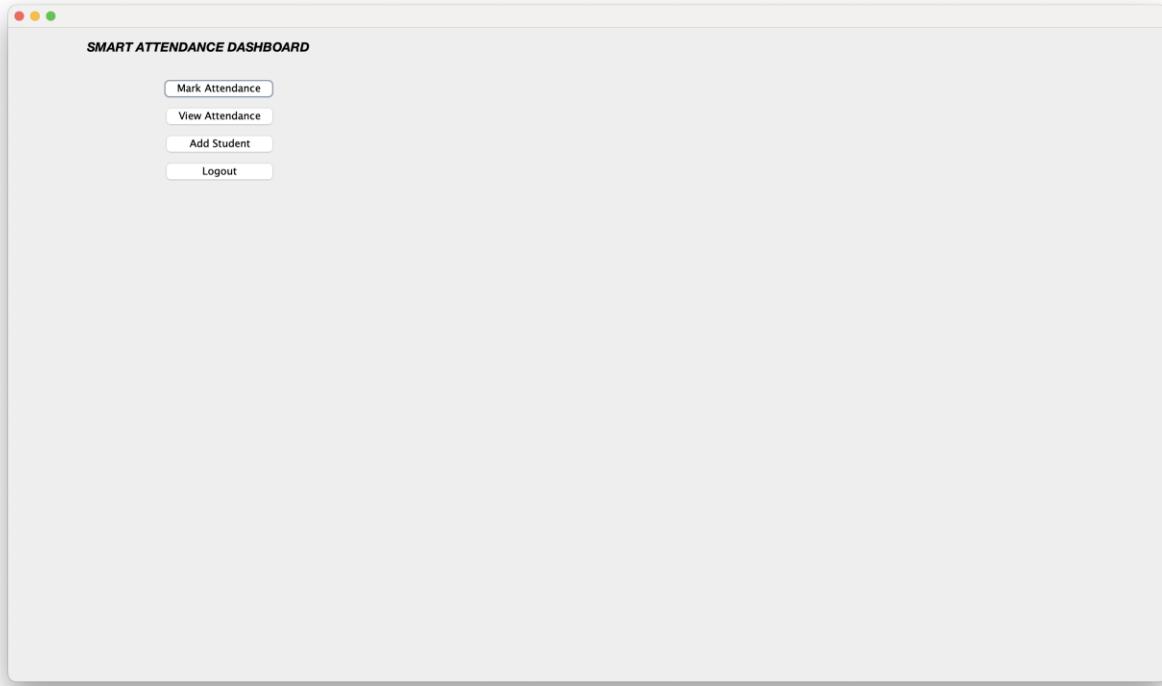
*/
try {
    for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex) {
    logger.log(java.util.logging.Level.SEVERE, null, ex);
}

//</editor-fold>

/*
 * Create and display the form
 */
java.awt.EventQueue.invokeLater(() -> new LoginFrame().setVisible(true));
}

// Variables declaration - do not modify
private javax.swing.JButton btnExit;
private javax.swing.JButton btnLogin;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JPasswordField txtPassword;
private javax.swing.JTextField txtUsername;
// End of variables declaration
}

```



This screenshot shows the main dashboard of the Smart Attendance System after successful login. It provides options such as marking attendance, viewing attendance records, adding new students, and logging out. The dashboard acts as the central control panel for accessing all system functionalities.

### CODE:

```
package smartattendance.gui;

import java.io.BufferedReader; i
import java.io.FileReader;
```

```

import java.io.IOException;

import javax.swing.JFrame; import smartattendance.logic.AttendanceManager; import
javax.swing.JOptionPane; import javax.swing.table.DefaultTableModel; import
smartattendance.gui.AttendanceFrame;

/** *

• @author navaneet.s

*/
public class DashboardFrame extends javax.swing.JFrame {

private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(DashboardFrame.class.getName());

/** *
* Creates new form DashboardFrame
*/
public DashboardFrame() {
    initComponents();
    setExtendedState(JFrame.MAXIMIZED_BOTH);
}

/**
* 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() {

jLabel1 = new javax.swing.JLabel();
btnMark = new javax.swing.JButton();
btnView = new javax.swing.JButton();
btnAdd = new javax.swing.JButton();

```



```

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
            .addComponent(btnLogout, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(btnAdd, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
            .addComponent(btnView, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))))))
        .addContainerGap(100, Short.MAX_VALUE))
    );
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(15, 15, 15)
        .addComponent(jLabel1)
        .addGap(29, 29, 29)
        .addComponent(btnMark)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnView)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnAdd)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(btnLogout)
        .addContainerGap(129, Short.MAX_VALUE))
    );
    pack();
}//
</editor-fold>
private void btnMarkActionPerformed(java.awt.event.ActionEvent evt) {

```

```

AttendanceFrame af = new AttendanceFrame(); af.setVisible(true); // show first
af.setExtendedState(JFrame.MAXIMIZED_BOTH); // then maximize this.dispose();

// TODO add your handling code here: }

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

ViewAttendanceFrame vf = new ViewAttendanceFrame(); vf.setVisible(true);
vf.setExtendedState(JFrame.MAXIMIZED_BOTH); this.dispose(); // TODO add your handling
code here: }

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

LoginFrame lf = new LoginFrame(); lf.setVisible(true); this.dispose(); // TODO add your
handling code here: }

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

new AddStudentFrame().setVisible(true);
this.dispose();

// 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.getInstalledLookAndFeels())
{
    if ("Nimbus".equals(info.getName()))
        javax.swing.UIManager.setLookAndFeel(info.getClassName());
    break;
}
}

} catch (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex)
{
    logger.log(java.util.logging.Level.SEVERE, null, ex);
}

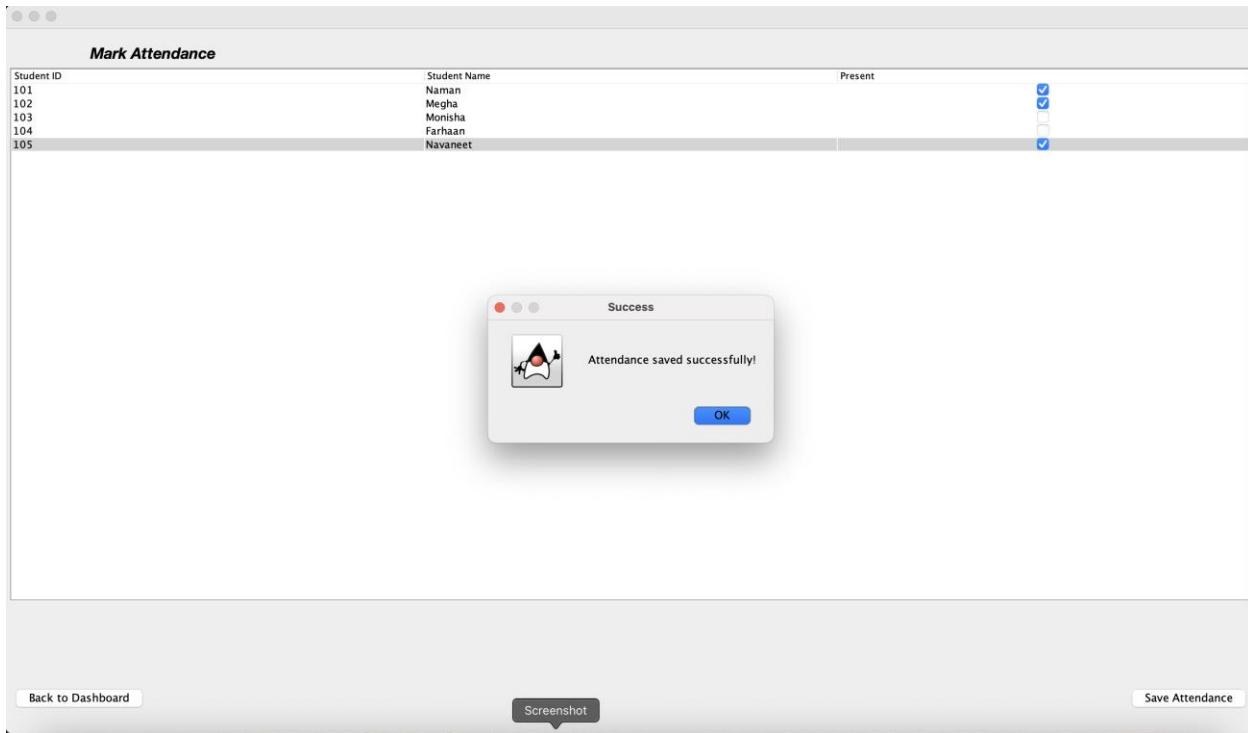
//</editor-fold>

/*
 * Create and display the form
 */
java.awt.EventQueue.invokeLater(() -> new DashboardFrame().setVisible(true));
}

// Variables declaration - do not modify
private javax.swing.JButton btnAdd;
private javax.swing.JButton btnLogout;
private javax.swing.JButton btnMark;
private javax.swing.JButton btnView;
private javax.swing.JLabel jLabel1;
// End of variables declaration
}

}

```



This screenshot shows the attendance marking interface where student details such as student ID and name are displayed in tabular form. The user can mark each student as present using checkboxes and save the attendance. The recorded data is then stored in the attendance records file for future reference.

**CODE:**

```

package smartattendance.gui; import java.io.BufferedReader; import java.io.FileReader; import
javax.swing.JFrame; import javax.swing.JOptionPane; import
javax.swing.table.DefaultTableModel; import java.io.FileWriter; import java.io.IOException;
import java.time.LocalDate; import javax.swing.JOptionPane; import
javax.swing.table.DefaultTableModel;

/** *

• @author navaneet.s

*/
public class AttendanceFrame extends javax.swing.JFrame {

private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(AttendanceFrame.class.getName());

/**
 * Creates new form AttendanceFrame
 */
public AttendanceFrame() {
    initComponents();
    pack(); // IMPORTANT
    setLocationRelativeTo(null); // center window
}

/**
 * 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() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
}

```

```

jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
jButton1 = new javax.swing.JButton();
jButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel2.setFont(new java.awt.Font("Helvetica Neue", 3, 18)); // NOI18N
jLabel2.setText("Mark Attendance");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
        {"101", "Naman", null},
        {"102", "Megha", null},
        {"103", "Monisha", null},
        {"104", "Farhaan", null},
        {"105", "Navaneet", null}
    },
    new String [] {
        "Student ID", "Student Name", "Present"
    }
))
Class[] types = {java.lang.String.class, java.lang.String.class, java.lang.Boolean.class};

public Class getColumnClass(int columnIndex) {
    return types [columnIndex];
}

jScrollPane1.setViewportView(jTable1);

jButton1.setText("Save Attendance");

```

```
jButton1.addActionListener(this::jButton1ActionPerformed);

jButton2.setText("Back")  
to  
Dashboard");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(136, 136, 136)
                .addComponent(jLabel1))
            .addGroup(layout.createSequentialGroup()
                .addGap(100, 100, 100)
                .addComponent(jLabel2)))
            .addGroup(layout.createSequentialGroup()
                .addGap(94, 94, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()
            .addGap(136, 136, 136)
            .addComponent(jButton2)
            .addGap(Short.MAX_VALUE)
            .addComponent(jButton1))
        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 0,
            Short.MAX_VALUE))))))
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        layout.createSequentialGroup()
            .addGap(Short.MAX_VALUE)
            .addComponent(jButton1))
        .addGroup(layout.createSequentialGroup()
            .addGap(Short.MAX_VALUE)
            .addComponent(jButton2)
            .addGap(136, 136, 136)
            .addComponent(jLabel1))
        .addGroup(layout.createSequentialGroup()
            .addGap(100, 100, 100)
            .addComponent(jLabel2))))
```

```

    .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jScrollPane1, javax.swing.GroupLayout.DEFAULT_SIZE, 152,
Short.MAX_VALUE)
        .addGap(100, 100, 100)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jButton1)
        .addComponent(jButton2))
        .addGap(27, 27, 27))
);

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

```

```

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
DashboardFrame db = new DashboardFrame();
db.setExtendedState(JFrame.MAXIMIZED_BOTH);
db.setVisible(true);

this.dispose(); // close attendance screen // TODO add your handling code here:
}

```

```

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

DefaultTableModel model = (DefaultTableModel) jTable1.getModel(); LocalDate today =
LocalDate.now();

try (FileWriter fw = new FileWriter("attendance_records.csv", true)) {

for (int i = 0; i < model.getRowCount(); i++) {

String studentId = model.getValueAt(i, 0).toString();
String studentName = model.getValueAt(i, 1).toString();

Boolean presentObj = (Boolean) model.getValueAt(i, 2);
boolean present = presentObj != null && presentObj;

String status = present ? "Present" : "Absent";

fw.write(today + "," + studentId + "," + studentName + "," + status + "\n");
}

JOptionPane.showMessageDialog(this,
    "Attendance saved successfully!",
    "Success",
    JOptionPane.INFORMATION_MESSAGE);

} catch (Exception e) {
JOptionPane.showMessageDialog(this,
    "Error saving attendance",
    "Error",
    JOptionPane.ERROR_MESSAGE);
e.printStackTrace();
} // 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.getInstalledLookAndFeels())  

            if ("Nimbus".equals(info.getName())){  

                javax.swing.UIManager.setLookAndFeel(info.getClassName());  

                break;  

            }  

        }  

    } catch (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex)  

    {  

        logger.log(java.util.logging.Level.SEVERE,      null,      ex);  

    }  

//</editor-fold>

/*      Create      and      display      the      form      */  

java.awt.EventQueue.invokeLater(()      ->      new      AttendanceFrame().setVisible(true));  

}

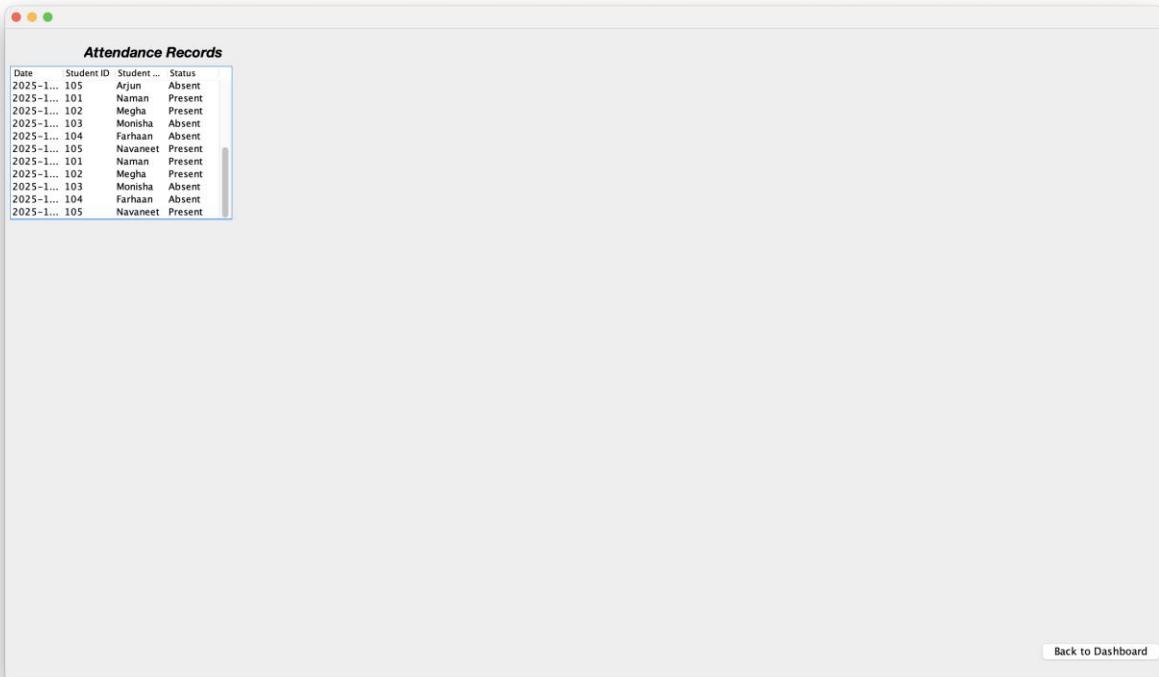
//      Variables      declaration      -      do      not      modify
private      javax.swing.JButton      jButton1;
private      javax.swing.JButton      jButton2;
private      javax.swing.JLabel      jLabel1;
```

```

private javax.swing.JLabel jLabel2;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

}

```



This screenshot displays the attendance records retrieved from the stored data file. It shows details such as date, student ID, student name, and attendance status. The system reads the attendance records and presents them in a structured tabular format for easy review.

**CODE:**

```

package smartattendance.gui;

/** *

• @author navaneet.s

*/ public class ViewAttendanceFrame extends javax.swing.JFrame {

private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(ViewAttendanceFrame.class.getName());

/** *
* Creates new form ViewAttendanceFrame
*/
public ViewAttendanceFrame() {
    initComponents();
    loadAttendanceFromCSV();
    setLocationRelativeTo(null);
}

/** *
* 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() {

jLabel1 = new javax.swing.JLabel();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
jButton1 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

jLabel1.setFont(new java.awt.Font("Helvetica Neue", 3, 18)); // NOI18N
jLabel1.setText("Attendance");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
new Object [][] {
{null, null, null, null, null},
{null, null, null, null, null},
{null, null, null, null, null},
{null, null, null, null, null}
},
new String [] {
"Date", "Student ID", "Student Name", "Status"
}
))
Class[] types = new Class []
{
}
);
}
}

```

```

        java.lang.String.class, java.lang.String.class, java.lang.String.class
    );
}

public Class getColumnClass(int columnIndex) {
    return types[columnIndex];
}

jScrollPane1.setViewportView(jTable1);

jButton1.setText("Back")  
to  
Dashboard");
jButton1.addActionListener(this::jButton1ActionPerformed);

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(100, 100, 100)
                .addComponent(jLabel1))
            .addGroup(layout.createSequentialGroup()
                .addGap(100, 100, 100)
                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 282,
                    javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(87, 87, Short.MAX_VALUE))
    .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
        layout.createSequentialGroup()
            .addGap(19, 19, 19)
            .addComponent(jLabel1)
            .addGap(195, 195, 195)
            .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(39, 39, 39)
            .addComponent(jButton1)
            .addGap(24, 24, 24)))
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1)
        .addGap(195, 195, 195)
        .addComponent(jButton1)
        .addGap(24, 24, 24)))
);
layout.setHorizontalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1)
        .addGap(195, 195, 195)
        .addComponent(jButton1)
        .addGap(24, 24, 24)))
);
layout.setVerticalGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(19, 19, 19)
        .addComponent(jLabel1)
        .addGap(195, 195, 195)
        .addComponent(jButton1)
        .addGap(24, 24, 24)))
);
}
}

private void loadAttendanceFromCSV() {
}

```

```

javax.swing.table.DefaultTableModel model = jTable1.getModel();
(model.setRowCount(0);

try (java.io.BufferedReader br = new java.io.BufferedReader(new java.io.FileReader("attendance_records.csv"))){
    String line;
    while ((line = br.readLine()) != null) {
        String[] data = line.split(",");
        model.addRow(data);
    }
} catch (Exception e) {
    javax.swing.JOptionPane.showMessageDialog(this,
        "No attendance records found");
}
}

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    DashboardFrame db = new DashboardFrame();
    db.setExtendedState(javax.swing.JFrame.MAXIMIZED_BOTH);
    db.setVisible(true);
    this.dispose(); // TODO add your handling code here:

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //
        /* 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 (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex) {
            logger.log(java.util.logging.Level.SEVERE, null, ex);
        }
    }
}

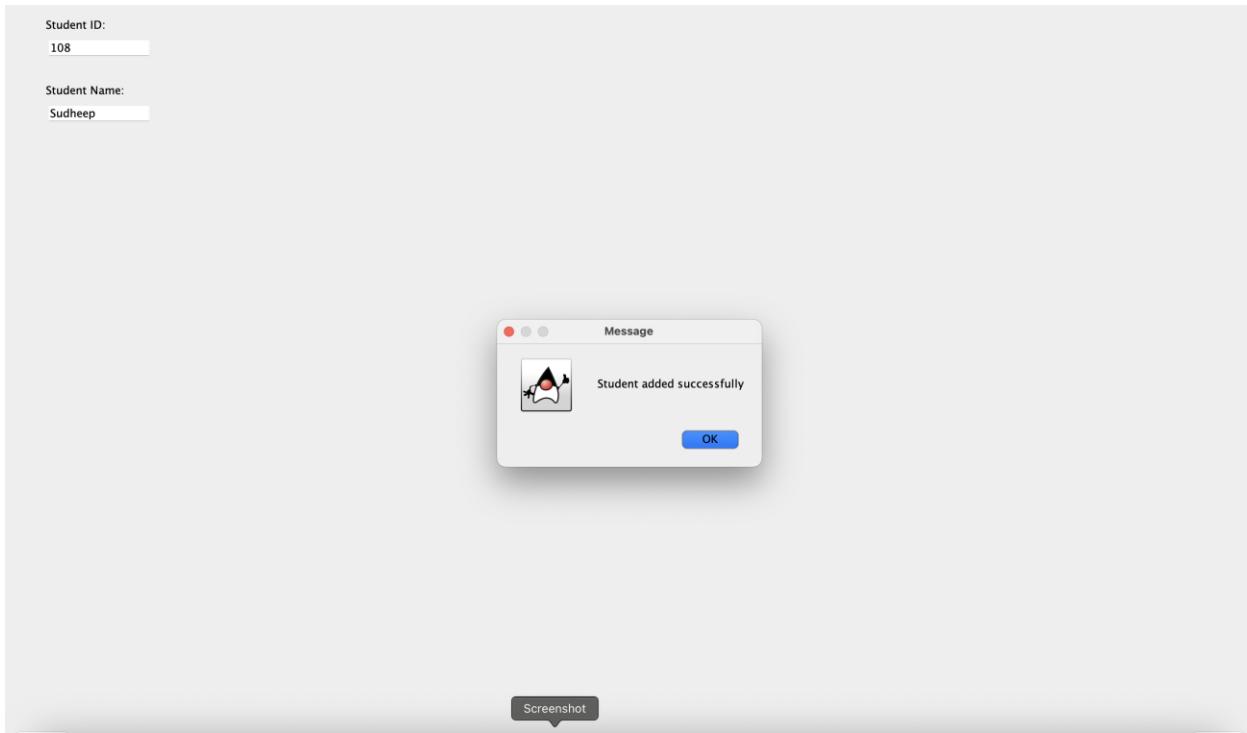
```

```
//</editor-fold>

/*
 * Create and display the form
 */
java.awt.EventQueue.invokeLater(() -> new ViewAttendanceFrame().setVisible(true));
}

// Variables declaration - do not modify
private javax.swing.JButton jButton1;
private javax.swing.JLabel jLabel1;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JTable jTable1;
// End of variables declaration
}

}
```



This screenshot shows the add student functionality where the user enters the student ID and student name. Upon submitting valid details, the system displays a confirmation message indicating successful addition. The student information is then stored in the student records file for future use.

### CODE:

```
package smartattendance.gui;

import java.io.FileWriter; import javax.swing.JOptionPane;
/** *
• @author navaneet.s

*/ public class AddStudentFrame extends javax.swing.JFrame {

private static final java.util.logging.Logger logger =
java.util.logging.Logger.getLogger(AddStudentFrame.class.getName());
```

```

/**
 * Creates new form AddStudentFrame
 */
public AddStudentFrame()
{
    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()
{
    jLabel1 = new javax.swing.JLabel();
    txtId = new javax.swing.JTextField();
    jLabel2 = new javax.swing.JLabel();
    txtName = new javax.swing.JTextField();
    btnSave = new javax.swing.JButton();
    btnBack = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

    jLabel1.setText("Student ID:");
    txtId.setText("ENTER");
    txtId.addActionListener(this::txtIdActionPerformed);

    jLabel2.setText("Student Name:");
    txtName.setText("Enter");
    txtName.addActionListener(this::txtNameActionPerformed);

    btnSave.setText("SAVE");
    btnSave.addActionListener(this::btnSaveActionPerformed);

    btnBack.setText("BACK");
    btnBack.addActionListener(this::btnBackActionPerformed);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(jLabel2)
                    .addComponent(txtName, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(49, 49, 49)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addComponent(btnSave)
                    .addComponent(btnBack)))
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(jLabel1)
                .addGap(49, 49, 49)
                .addComponent(txtId, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)))
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(jLabel1)
                .addGap(49, 49, 49)
                .addComponent(txtId, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(jLabel2)
                .addGap(49, 49, 49)
                .addComponent(txtName, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(btnSave))
            .addGroup(layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(btnBack)))
    );
}

```

```

        .addComponent(txtId)
        .addComponent(txtName,      javax.swing.GroupLayout.DEFAULT_SIZE,      125,
Short.MAX_VALUE))
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,  Short.MAX_VALUE))
.addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(btnBack)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,      244,
Short.MAX_VALUE)
        .addComponent(btnSave)
        .addContainerGap())
);
layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(18,                  18, 18)
        .addComponent(jLabel1)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtId,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,  javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(29,                  29, 29)
        .addComponent(jLabel2)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(txtName,          javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,  javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,      102,
Short.MAX_VALUE)
    )
    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(btnBack)
        .addComponent(btnSave))
        .addGap(36,                  36, 36))
);
pack();
}//</editor-fold>

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

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

this.dispose(); new DashboardFrame().setVisible(true); // TODO add your handling code here: }

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

String id = txtId.getText().trim(); String name = txtName.getText().trim();
```

```

if (id.isEmpty() || name.isEmpty()) {
    JOptionPane.showMessageDialog(this, "All fields are required");
    return;
}

try {
    FileWriter fw = new FileWriter("students.csv", true);
    fw.write(id + "," + name + "\n");
    fw.close();

    JOptionPane.showMessageDialog(this, "Student added successfully");

    txtId.setText("");
    txtName.setText("");
}

} catch (Exception e) {
    JOptionPane.showMessageDialog(this, "Error saving student");
}
// TODO add your handling code here:
}

private void txtNameActionPerformed(java.awt.event.ActionEvent 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.getInstalledLookAndFeels()) {
        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;
        }
    }
} catch (ReflectiveOperationException | javax.swing.UnsupportedLookAndFeelException ex) {
    logger.log(java.util.logging.Level.SEVERE, null, ex);
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(() -> new AddStudentFrame().setVisible(true));
}

```

```
//      Variables      declaration      -      do      not      modify
private          javax.swing.JButton      btnBack;
private          javax.swing.JButton      btnSave;
private          javax.swing.JLabel       jLabel1;
private          javax.swing.JLabel       jLabel2;
private          javax.swing.JTextField    txtId;
private          javax.swing.JTextField    txtName;
//      End      of      variables      declaration
}
}
```

## 9. NEW LEARNINGS FROM THE PROJECT

Through this project, the following learnings were achieved:

- Understanding of real-world problem solving using Java
- Practical usage of file handling and CSV files
- Improved knowledge of NetBeans IDE
- Experience in modular programming
- Understanding how data persistence works without databases
- Improved debugging and logical thinking skills

## **10. FUTURE ENHANCEMENTS**

The project can be enhanced further by:

- Adding a database (MySQL) instead of CSV files
- Implementing user authentication (login system)
- Generating attendance reports automatically
- Adding GUI enhancements
- Integrating date-wise and subject-wise attendance

- Exporting attendance in PDF format

## **References :**

1. Herbert Schildt, *Java: The Complete Reference*, McGraw-Hill.
2. Y. Daniel Liang, *Introduction to Java Programming*, Pearson.
3. Oracle Java Documentation – Java SE and Swing Tutorials.
4. Class notes and course materials.
5. Other AI tools.

