CRIMINAL RECORD MANAGEMENT

A MINI-PROJECT BY:

SIVA BHARATHI K 230701317

SOWBARNIGAA SRIDHARAN 230701327

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **“CRIMINAL RECORD MANAGEMENT”** is the bonafide work of **“SIVA BHARATHI K , SOWBARNIGAA SRIDHARAN”** who carried out theproject work under my supervision.

Submitted for the practical examination held on ______________

SIGNATURE

Mr.G SARAVANA GOKUL
Assistant Professor (SS),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam, Chennai-602105

SIGNATURE

Ms.V.JANANEE
Assistant Professor (SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam, Chennai-602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

The Criminal Record Management System is a Java-based mini-project-which helps to modernize and automate the management of Criminal records. Our application facilitates the management of sensitive data in such a way that it scales up the accuracy and efficiency of the task, while at the same time One eliminates the manual coping of records. The system aims to keep detailed records pertaining to each criminal: personal profiles; crime histories; case statuses of offenders-into a central and secure database.

The application is based on Java with database integration to carry out uninterrupted data storage and retrieval. It allows for adding, displaying or deleting entries by the authorized personnel with complete data security through user authentication. The system provides exhaustive search and filtering facility to enable faster retrieval of records satisfying certain conditions. The system also helps generate detailed reports, which would help the law enforcement agencies to analyze trends and manage cases effectively.

With hands-on on modern programming techniques, along with intuitive interaction design, the project exemplifies how technology can radically restore record keeping in the area of criminal justice. This provides a prototype design toward scalable real-world applications to improve administrative efficiency without resulting in an error-free repository of critical data.

TABLE OF CONTENTS

INTRODUCTION

1.1 INTRODUCTION

Criminal Record Management System is a Java-based application designed to efficiently manage and track criminal records. This system aims to streamline the process of recording, storing, and retrieving information related to crimes, criminals, and investigations.

1.2 IMPLEMENTATION

The **CRIMINAL RECORD MANAGEMENT** project discussed here is implemented usingthe concepts of **JAVA SWINGS** and **MYSQL**.

1.3 SCOPE OF THE PROJECT

This project offers significant potential in transforming how criminal records are handled. Its primary scope lies in providing an efficient, secure, and user-friendly platform for managing criminal data in a centralized database. This project caters to the needs of law enforcement agencies, judicial bodies, and administrative staff who require a reliable system to store, retrieve, and update sensitive information about individuals involved in criminal activities.

SYSTEM SPECIFICATIONS

2.1 HARDWARE SPECIFICATIONS:

PROCESSOR	:	Intel i5
MEMORY SIZE	:	4GB(Minimum)
HARD DISK	:	500 GB of free space

2.2 SOFTWARE SPECIFICATIONS:

PROGRAMMING LANGUAGE	:	Java, MySQL
FRONT-END	:	Java Swing
BACK-END	:	MySQL
OPERATING SYSTEM	:	Windows 10

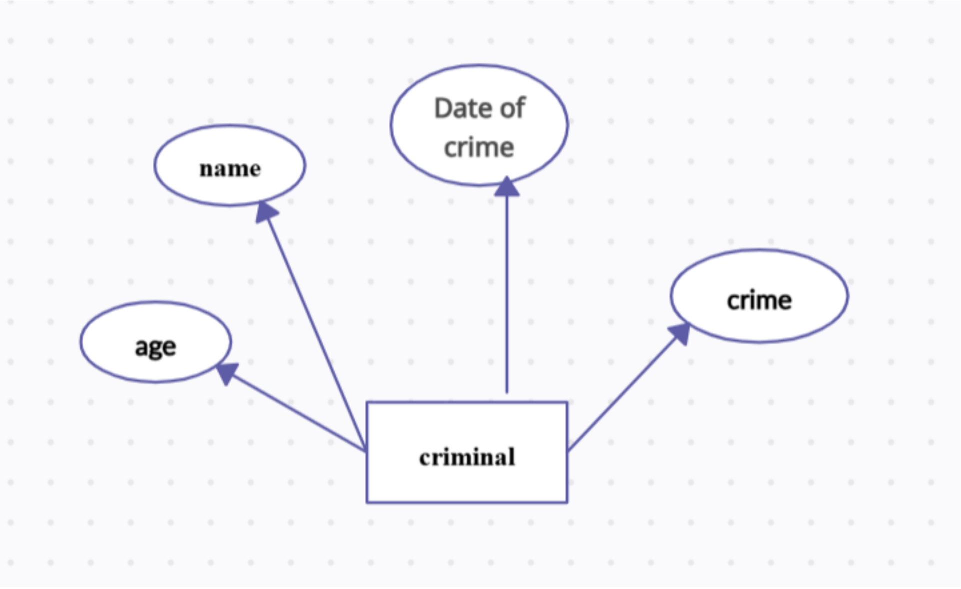
ENTITY RELATION MODEL

3.1 ER DIAGRAM

An Entity Relationship diagram describes the structure of the database.

In this Crime Record Management System Admin will be able to login and be able to add criminal, view all the criminal details and also can delete criminal records.

In this simplified model, there are no explicit relationships between the **Criminal** entity and other entities. All the information about the crime, age, name, and date is directly associated with the criminal record.



SAMPLE CODE

4.1 LOGIN PAGE BACKEND:

```
  // Create and show the login screen
JFrame loginFrame = new JFrame("Login");
loginFrame.setSize(300, 200);
loginFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
loginFrame.setLocationRelativeTo(null);

JPanel loginPanel = new JPanel(new GridLayout(3, 2));
loginPanel.add(new JLabel("Username:"));
JTextField usernameField = new JTextField();
loginPanel.add(usernameField);

loginPanel.add(new JLabel("Password:"));
JPasswordField passwordField = new JPasswordField();
loginPanel.add(passwordField);

JButton loginButton = new JButton("Login");
loginPanel.add(loginButton);

loginFrame.add(loginPanel);
loginFrame.setVisible(true);

loginButton.addActionListener(new ActionListener() {
@Override
public void actionPerformed(ActionEvent e) {
        String username = usernameField.getText();
        String password = new String(passwordField.getPassword());
       //Simple username/password check
```

```
                if (username.equals("admin") && password.equals("password")) {
                    loginFrame.dispose(); // Close login frame
                    SwingUtilities.invokeLater(() -> {
                        CriminalManagementSystemUI ui = new CriminalManagementSystemUI();
                        ui.setVisible(true);
                    });
                } else {
                        JOptionPane.showMessageDialog(loginFrame, "Invalid credentials. Please try
again.");
                }
            }
        });
    }
}
```

4.2 SOURCE CODE:

```
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.ArrayList;
import java.util.Iterator;
import java.util.List;

public class CriminalManagementSystemUI extends JFrame {
    private CriminalManagement criminalManagement;
    private JTextField nameField, ageField, crimeField, dateOfCrimeField, searchField;
    private JTable criminalTable;
    private DefaultTableModel tableModel;

    public CriminalManagementSystemUI() {
        criminalManagement = new CriminalManagement();
        setTitle("Criminal Management System");
        setSize(600, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
```

```
// Create tabs
 JTabbedPane tabbedPane = new JTabbedPane();

 // Create the Add Criminal panel
 JPanel addCriminalPanel = new JPanel(new GridLayout(5, 2));
 addCriminalPanel.add(new JLabel("Name:"));
 nameField = new JTextField();
 addCriminalPanel.add(nameField);

 addCriminalPanel.add(new JLabel("Age:"));
 ageField = new JTextField();
 addCriminalPanel.add(ageField);

 addCriminalPanel.add(new JLabel("Crime:"));
 crimeField = new JTextField();
 addCriminalPanel.add(crimeField);

 addCriminalPanel.add(new JLabel("Date of Crime:"));
 dateOfCrimeField = new JTextField();
 addCriminalPanel.add(dateOfCrimeField);

 JButton addButton = new JButton("Add Criminal");
 addCriminalPanel.add(addButton);

 tabbedPane.addTab("Add Criminal", addCriminalPanel);

 // Create the Display Criminal panel
 String[] columnNames = {"Name", "Age", "Crime", "Date of Crime"};
 tableModel = new DefaultTableModel(columnNames, 0);
 criminalTable = new JTable(tableModel);
 JScrollPane scrollPane = new JScrollPane(criminalTable);
 tabbedPane.addTab("Display Criminals", scrollPane);

 // Create the Delete Criminal panel
 JPanel modifyDeletePanel = new JPanel(new GridLayout(2, 2));
 modifyDeletePanel.add(new JLabel("Search Name:"));
 searchField = new JTextField();
 modifyDeletePanel.add(searchField);

 JButton deleteButton = new JButton("Delete Criminal");
 modifyDeletePanel.add(deleteButton);

 tabbedPane.addTab("Delete Criminal", modifyDeletePanel);
```

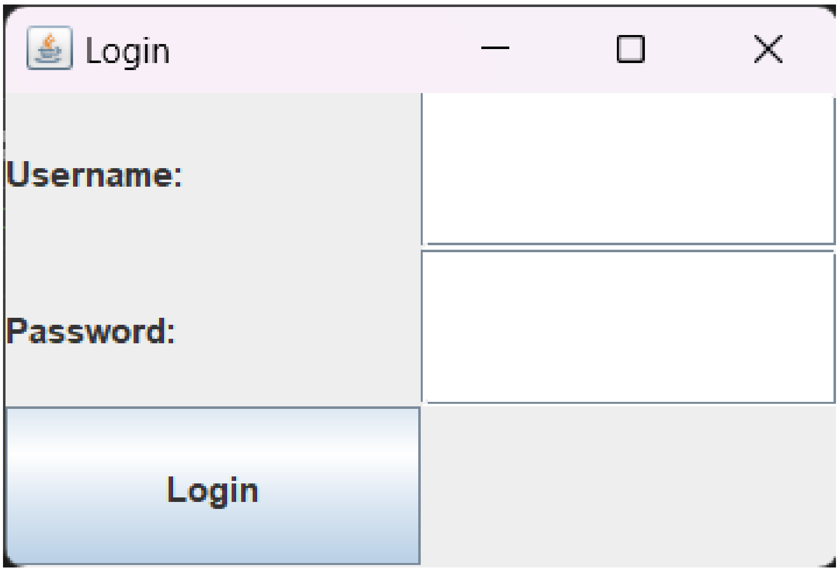
```
// Add everything to the frame
 add(tabbedPane);

 // Button actions
 addButton.addActionListener(new ActionListener() {
     @Override
     public void actionPerformed(ActionEvent e) {
         try {
             // Get input values
             String name = nameField.getText();
             int age = Integer.parseInt(ageField.getText());
             String crime = crimeField.getText();
             String dateOfCrime = dateOfCrimeField.getText();

             // Create new criminal and add to the system
             Criminal criminal = new Criminal(name, age, crime, dateOfCrime);
             criminalManagement.addCriminal(criminal);

             // Clear form fields
             nameField.setText("");
             ageField.setText("");
             crimeField.setText("");
             dateOfCrimeField.setText("");

             // Update the table
             updateCriminalTable();
         } catch (NumberFormatException ex) {
             JOptionPane.showMessageDialog(null, "Please enter a valid age.");
         }
     }
 });

 deleteButton.addActionListener(new ActionListener() {
     @Override
     public void actionPerformed(ActionEvent e) {
         // Get the name to delete
         String nameToDelete = searchField.getText().trim();
         if (!nameToDelete.isEmpty()) {
             boolean deleted = criminalManagement.deleteCriminalByName(nameToDelete);
             if (deleted) {
                 JOptionPane.showMessageDialog(null, "Criminal deleted successfully.");
             } else {
                 JOptionPane.showMessageDialog(null, "Criminal not found.");
             }
```

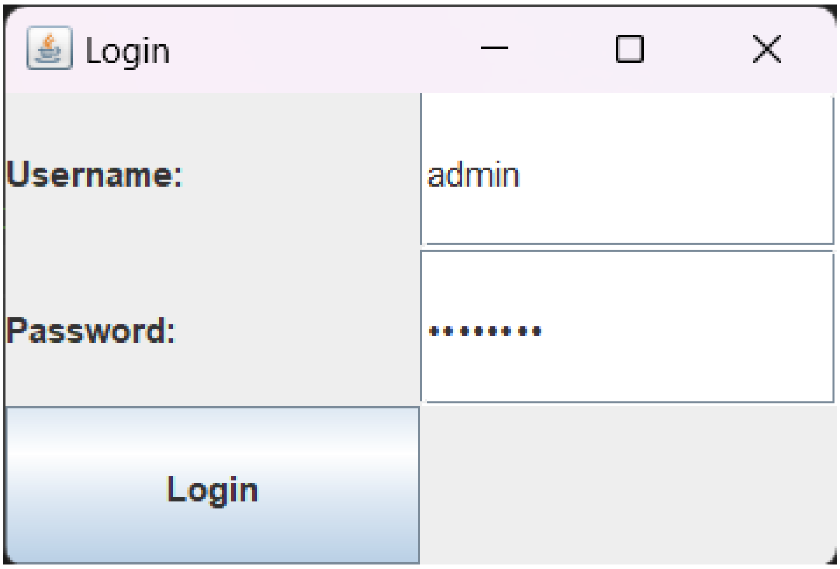
```
                    // Clear the input field
                    searchField.setText("");

                    // Update the table
                    updateCriminalTable();
                } else {
                    JOptionPane.showMessageDialog(null, "Please enter a name to delete.");
                }
            }
        });
    }

    private void updateCriminalTable() {
        // Clear the table
        tableModel.setRowCount(0);
        List<Criminal> criminals = criminalManagement.getAllCriminals();
        for (Criminal criminal : criminals) {
            tableModel.addRow(new Object[]{criminal.getName(), criminal.getAge(),
criminal.getCrime(), criminal.getDateOfCrime()});
        }
    }

    public static void main(String[] args) {
        // Create and show the login screen
        JFrame loginFrame = new JFrame("Login");
        loginFrame.setSize(300, 200);
        loginFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        loginFrame.setLocationRelativeTo(null);

        JPanel loginPanel = new JPanel(new GridLayout(3, 2));
        loginPanel.add(new JLabel("Username:"));
        JTextField usernameField = new JTextField();
        loginPanel.add(usernameField);

        loginPanel.add(new JLabel("Password:"));
        JPasswordField passwordField = new JPasswordField();
        loginPanel.add(passwordField);

        JButton loginButton = new JButton("Login");
        loginPanel.add(loginButton);

        loginFrame.add(loginPanel);
        loginFrame.setVisible(true);

        loginButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());

                // Simple username/password check
                if (username.equals("admin") && password.equals("password")) {
                    loginFrame.dispose(); // Close login frame
                    SwingUtilities.invokeLater(() -> {
                        CriminalManagementSystemUI ui = new CriminalManagementSystemUI();
                        ui.setVisible(true);
                    });
                } else {
                        JOptionPane.showMessageDialog(loginFrame, "Invalid credentials. Please try
again.");
                }
            }
        });
    }
}

class Criminal {
    private String name;
    private int age;
    private String crime;
    private String dateOfCrime;

    public Criminal(String name, int age, String crime, String dateOfCrime) {
        this.name = name;
        this.age = age;
        this.crime = crime;
        this.dateOfCrime = dateOfCrime;
    }
    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getCrime() {
        return crime;
    }
```

```
   public String getDateOfCrime() {
        return dateOfCrime;
    }
}

class CriminalManagement {
    private List<Criminal> criminals;

    public CriminalManagement() {
        criminals = new ArrayList<>();
    }

    public void addCriminal(Criminal criminal) {
        criminals.add(criminal);
    }
   public List<Criminal> getAllCriminals() {
        return criminals;
    }

    public boolean deleteCriminalByName(String name) {
        Iterator<Criminal> iterator = criminals.iterator(); // Use an iterator
        while (iterator.hasNext()) {
            Criminal criminal = iterator.next();
            if (criminal.getName().equalsIgnoreCase(name)) { // Case-insensitive search
                iterator.remove(); // Safe removal
                return true; // Criminal deleted
            }
        }
        return false; // Criminal not found
    }
}
```

SNAPSHOTS

5.1 LOGIN PAGE:

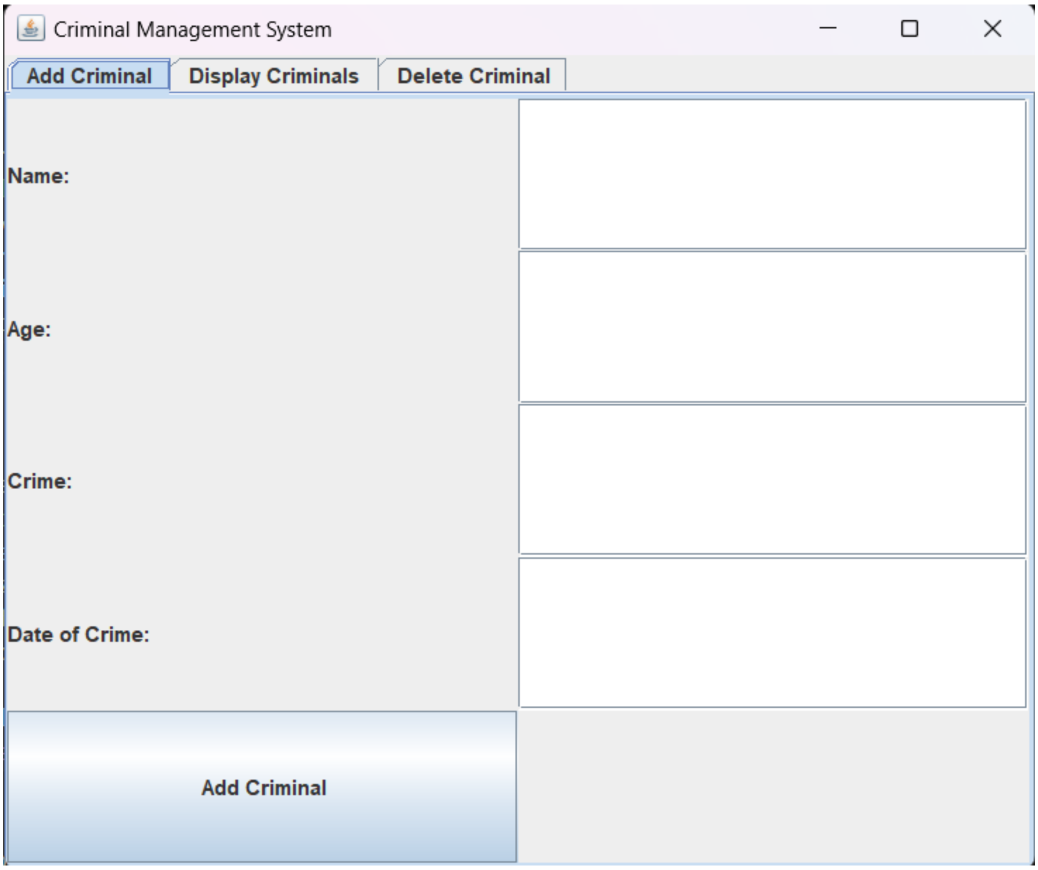


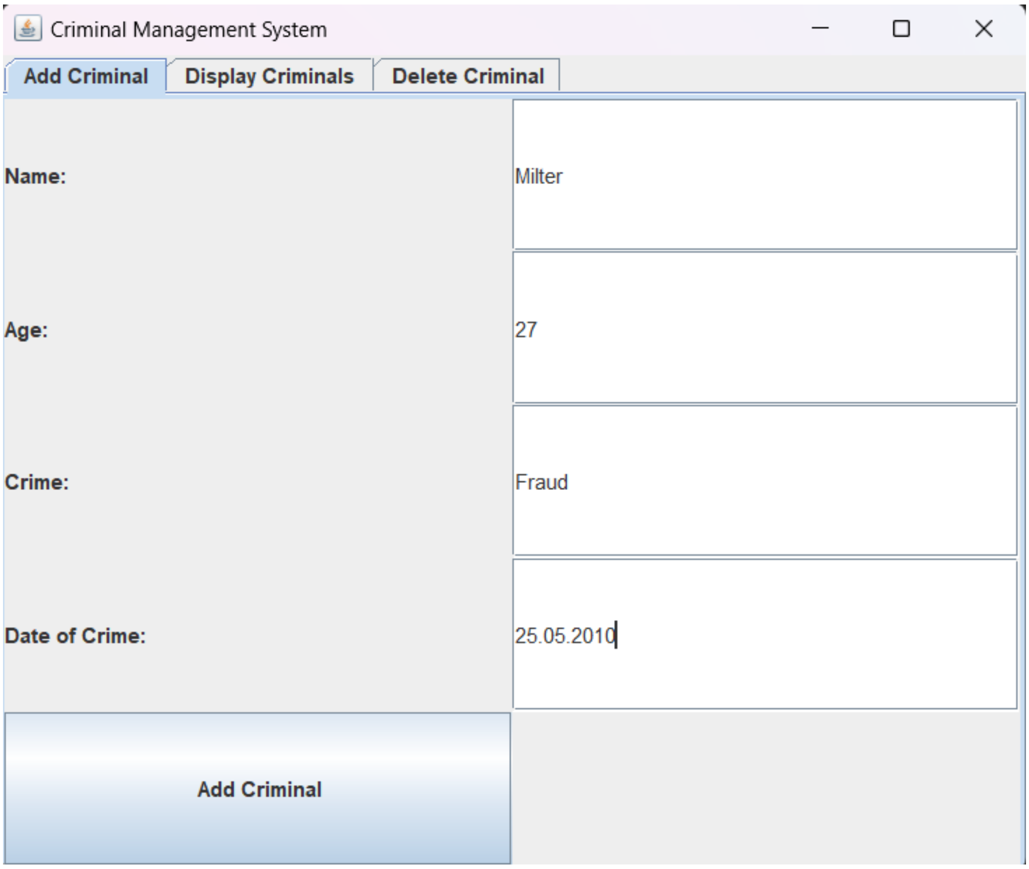


Username: admin

Password: ••••••••

5.2 ADD CRIMINAL PAGE





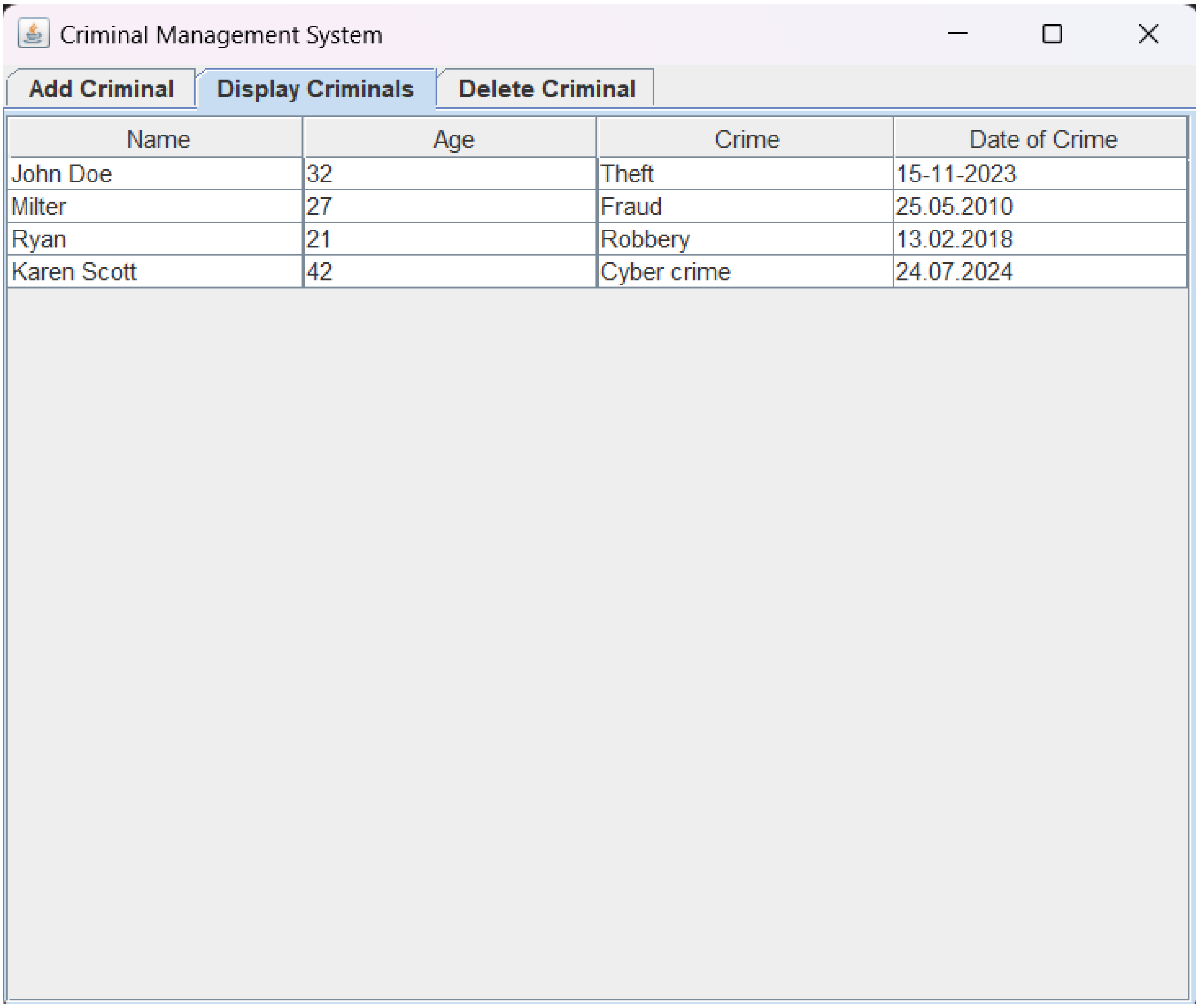
Name: Milter

Age: 27

Crime: Fraud

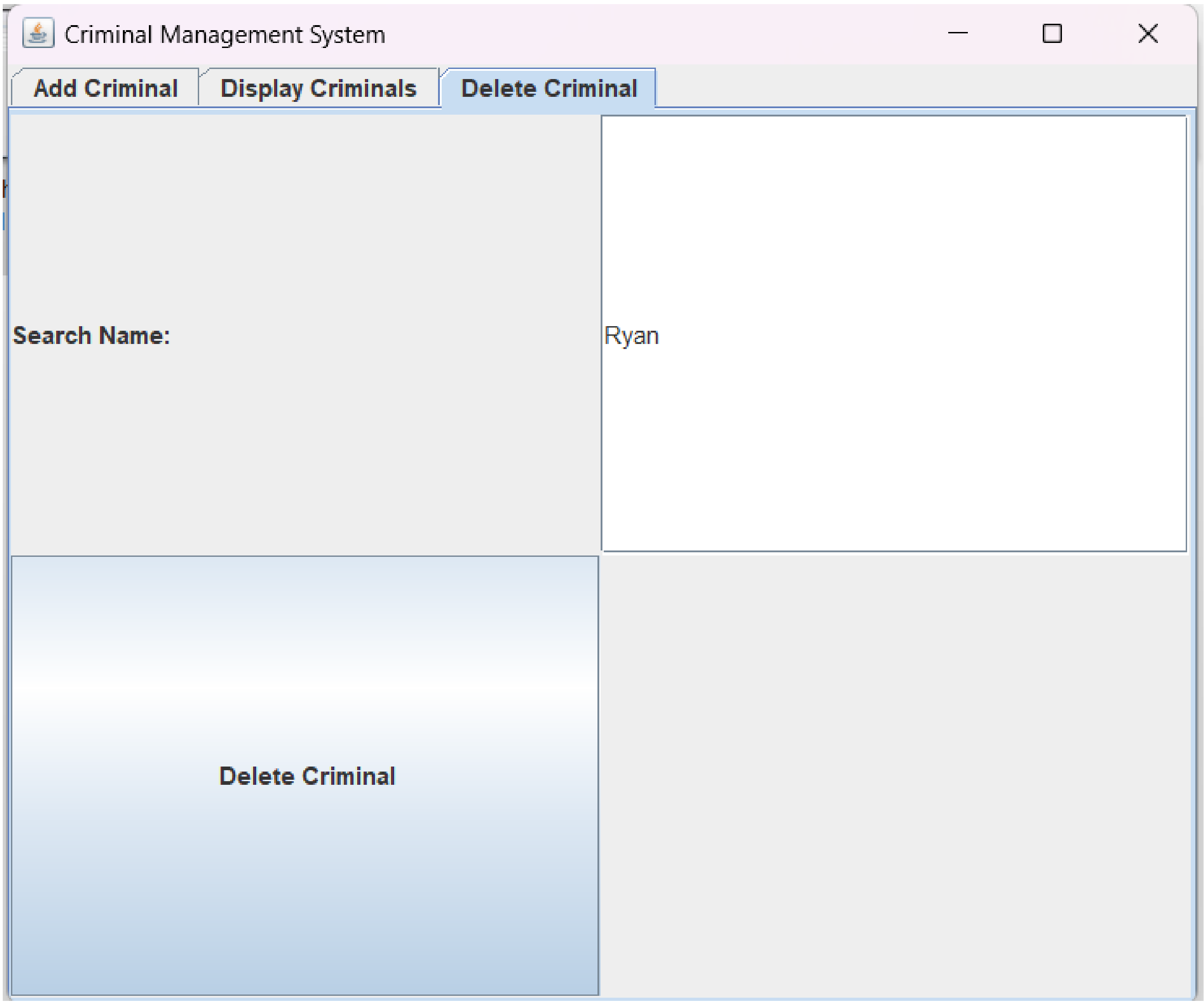
Date of Crime: 25.05.2010

5.3 DISPLAY CRIMINAL PAGE



Name	Age	Crime	Date of Crime
John Doe	32	Theft	15-11-2023
Milter	27	Fraud	25.05.2010
Ryan	21	Robbery	13.02.2018
Karen Scott	42	Cyber crime	24.07.2024

5.4 DELETE CRIMINAL PAGE



Search Name: Ryan

CONCLUSION

The project, offers a comprehensive solution for efficient management of criminal records. By automating various tasks, this system can significantly improve the accuracy and efficiency of law enforcement agencies. It provides a centralized platform for storing, retrieving, and analyzing criminal data, facilitating informed decision-making and effective crime prevention. It provides a user-friendly interface for data input and retrieval.

REFERENCES

1. **https://www.javatpoint.com/java-tutorial**

2. https://www.wikipedia.org/

3. https://www.w3schools.com/sql/

4. SQL | Codecademy