

**ONLINE BOOK TRACKER**

**CS23333-OBJECT ORIENTED PROGRAMMING USING JAVA**

*Submitted by*

DHUVARAGESH H B -231001037

HARI VIGNESH V-231001051

*Of*

**BACHELOR OF TECHNOLOGY**

*In*

**INFORMATION TECHNOLOGY**

**RAJALAKSHMI ENGINEERING COLLEGE, THANDALAM**  
**(An Autonomous Institution)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**RAJALAKSHMI ENGINEERING COLLEGE**

**THANDALAM ,CHENNAI 600 025**

**NOVEMBER 2024**

## **BONAFIDE CERTIFICATE**

Certified that this project report titled “**ONLINE BOOK TRACKER**” is the Bonafide work of **DHUVARAGESH H B(231001037), HARI VIGNESH V(231001051)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**SIGNATURE**

**Dr. P. Valarmathie**

**HEAD OF THE DEPARTMENT**

Information Technology  
Rajalakshmi Engineering College,  
Rajalakshmi Nagar, Thandalam  
Chennai – 602105

**SIGNATURE**

**Mrs. Usha S**

**COURSE INCHARGE**

Information Technology  
Rajalakshmi Engineering College  
Rajalakshmi Nagar, Thandalam  
Chennai – 602105

This project is submitted for IT19341 – Introduction to Oops and Java held on

---

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## 1.1 Abstract

The Book Tracker project is a visionary initiative designed to transform the way individuals manage their book collections in the digital age. With the increasing shift in reading habits and the growing reliance on digital tools, the need for a comprehensive, user-friendly platform to organize and track books has become more apparent. This project seeks to address this need by providing an intuitive, efficient, and scalable solution that bridges traditional methods of book management with the convenience of modern technology.

At its core, the Book Tracker offers users the ability to add, update, delete, and search for books in their personal collections. Each entry can be enriched with detailed information, including the title, author, and edition, allowing users to maintain a precise and well-organized literary record. By incorporating a powerful search function, the platform eliminates the challenge of sorting through large collections, enabling users to locate books effortlessly based on specific criteria. The update and delete features further ensure that the records remain current and relevant, making it easy to manage even extensive archives with minimal effort.

Technically, the platform is built with a robust backend powered by Java, ensuring reliable performance and seamless operations. MongoDB is utilized for database storage, providing scalable and secure management of book records, accommodating users' growing libraries without compromising efficiency. This combination of technologies ensures a strong foundation for handling the diverse needs of readers, from casual enthusiasts to serious bibliophiles.

The Book Tracker is more than just a storage solution; it is envisioned as a resource that enhances users' reading journeys by enabling them to interact with their collections in meaningful ways. By offering a personalized and accessible experience, the platform empowers users to set and achieve reading goals, explore their literary interests, and stay connected to their love for books. Whether managing a small personal library or an expansive collection, the Book Tracker simplifies the process and fosters a deeper appreciation for literature.

Ultimately, this project aspires to redefine book collection management by combining traditional values of organization and care with the benefits of digital innovation. It provides an all-encompassing solution that is intuitive, scalable, and impactful, ensuring that readers can maintain their collections with ease and efficiency anytime, anywhere. The Book Tracker serves as a testament to the potential of technology to enhance everyday activities, making it an invaluable tool for readers in the 21st century.

## TABLE OF CONTENTS

### 1. ONLINE BOOK TRACKING SYSTEM

1.1. Abstract-----	5
1.2. Introduction -----	5
1.3. Purpose -----	5
1.4. Scope of Project-----	6
1.5. Software Requirement Specification -----	6
2. Product Fuctions -----	11
2.1. Sigificance And Benefits -----	11
2.2. Use Management Module -----	11
3. Design-----	13
4. Implementation-----	15
5. Conclusion -----	60

## ACKNOWLEDGEMENT

First, we thank the almighty God for the successful completion of the project. Our sincere thanks to our chairman **Mr.S. Meganathan, B.E., F.I.E** for his sincere endeavour in educating us in his premier institution. We would like to express our deep gratitude to our beloved Chairperson **Dr.Thangam Meganathan**, for her enthusiastic motivation which inspired us a lot in completing this project and Vice-Chairman **Mr. Abhay Shankar Meganathan B.E., M.S.**, for providing us with the requisite infrastructure.

We also express our sincere gratitude to our college principal, **Dr.S.N.Murugesan M.E., PhD.**, for his kind support and facilities to complete our work on time. We extend heartfelt gratitude to **Dr.P.Valarmathie, Professor and Head of the Department of Information Technology** for her guidance and encouragement throughout the work. We are very glad to thank our course faculty **Mrs. Usha S, Professor** of our department for their encouragement and support towards the successful completion of this project. We extend our thanks to our parents, friends, all faculty members, and supporting staff for their direct and indirect involvement in the successful completion of the project for their encouragement and support.

**DHUVARAGESH H B**  
**HARI VIGNESH V**

## 1.2 Introduction

### **Welcome to the Book Tracker Website!**

Welcome to a revolutionary way to organize and manage your literary world! The Book Tracker website is a cutting-edge, user-friendly platform designed to simplify the process of cataloging, storing, and tracking your book collection. Whether you are an avid reader with a vast library or someone just starting to build a collection, this platform is tailored to meet your needs and enrich your reading journey.

This online library management tool combines intuitive design with robust technology, offering a seamless experience for organizing your books. Built with Java, the platform ensures reliable and efficient performance, while MongoDB provides dynamic and scalable storage capabilities. Together, these technologies create a powerful backend infrastructure, ensuring that the system evolves alongside your collection as it grows in size and complexity. No matter how expansive or diverse your library becomes, the Book Tracker is ready to adapt.

The Book Tracker allows you to personalize your collection by adding detailed information about each book, including its title, author, and edition. This makes it easy to organize your library in a way that suits your preferences, whether you're curating a collection of timeless classics, tracking your favorite book series, or creating a digital archive of rare editions. With features like advanced search functionality, you can quickly locate any book in your collection, saving time and eliminating the frustration of searching through stacks of books or digital files.

Beyond its core organizational features, the platform is designed to enhance your reading experience. By offering a dedicated space to record and track your progress, the Book Tracker helps you set and achieve your reading goals. Whether you're revisiting old favorites or exploring new genres, this tool empowers you to stay connected with your literary interests in a meaningful way.

In addition to being a powerful tool for personal use, the Book Tracker is built to accommodate a wide range of collections, from small private libraries to extensive archives. Its scalable design ensures that it remains fast and efficient, no matter how large your library grows. The ability to update and delete book entries further ensures that your collection stays relevant and up-to-date.

Experience the convenience of having your entire library at your fingertips. With the Book Tracker, you can explore endless possibilities for managing your books, maintain a well-organized library, and immerse yourself in a streamlined, tech-driven reading experience. Embrace the future of book management and take the first step toward transforming how you interact with your literary world. Whether for personal growth, academic pursuits, or pure enjoyment, the Book Tracker is here to redefine the way you manage and cherish your books.

### 1.3 Purpose

The primary purpose of this project is to create an innovative, user-friendly digital platform that redefines how individuals manage their book collections. As reading habits evolve in the digital age, the need for a comprehensive, accessible, and personalized system for organizing books has grown significantly. This Book Tracker addresses that need by providing a seamless online solution for storing and maintaining literary records, helping users keep track of their reading journey in a structured and efficient way.

By leveraging the capabilities of Java for robust backend functionality and MongoDB for scalable, secure storage, the project aims to deliver a reliable tool capable of handling a diverse range of book collections, from small personal libraries to extensive archives. Beyond simple storage, this tracker is envisioned as a resource that empowers users to interact with their collections in meaningful ways, ensuring they remain engaged with their reading goals and interests.

Ultimately, this project aspires to bridge the gap between traditional book collection methods and modern digital convenience, fostering a deeper appreciation for literature while offering an easy-to-use, tech-driven solution for readers in the 21st century.

### 1.4 Scope of the Project

The scope of the **Online book tracker** encompasses a range of functionalities designed to enhance the educational process by leveraging data management and analysis tools. The system aims to streamline the process of recording, storing, and analyzing student marks. It provides educators with a robust platform to input student marks, calculate various statistics, and generate comprehensive performance reports.

One of the core components of this project is its integration with a SQLite database using JDBC (Java Database Connectivity). JDBC enables seamless interaction between the application and the SQLite database, allowing for efficient data storage and retrieval. This connectivity ensures that student marks are securely stored and can be accessed for real-time analysis. The system supports multiple features such as the insertion of student marks, calculation of statistical measures like mean and standard deviation, and the identification of top and lowest performers. Additionally, it offers a user-friendly interface for educators to view and analyze student performance data, facilitating informed decision-making and targeted educational interventions.

## 1.5 Software Requirement Specification

### 1. Backend Software

#### 1. Java (Spring Boot Framework)

- **Role:**

- Handles server-side logic for user authentication, book management operations (add, update, delete, search), and communication with the database.
- Provides RESTful APIs for interaction between the frontend and backend.
- Robust and scalable framework.
- Integration with MongoDB through libraries like Spring Data MongoDB.
- High performance and security features.

## 2. Database

### 2. MongoDB

- **Role:**
  - Stores book records, user data, and other app-related information in a flexible, document-oriented format.
  - Manages dynamic data attributes for books (e.g., title, author, genre, edition).
  - Provides indexing for fast data retrieval and supports replication for high availability.
- **Why Used:**
  - Flexible schema for accommodating varying book attributes.
  - Scalability to handle large collections and user growth.
  - Excellent performance for read and write operations.

## 3. Frontend Software

### 3. React/Angular (Frontend Framework)

- **Role:**
  - Builds the user interface (UI) for the Book Tracker application, allowing users to interact with the platform.
  - Displays forms for adding/updating books, dashboards for analytics, and search functionalities.
  - Communicates with the backend using RESTful APIs.
- **Why Used:**
  - React offers component-based architecture for reusability and dynamic rendering.
  - Angular provides a structured, two-way data-binding framework for complex interfaces.
  - Both ensure responsive and interactive design.

## PRODUCT FUNCTION

### Add a Book

The "Add a Book" feature lets you easily add books to your collection by entering details like title, author, and edition. It ensures every book is well-organized in your digital library, making the process quick and hassle-free.

### Find a Book

The "Find a Book" feature allows you to quickly locate any book in your collection by searching for its title, author, or edition. The smart search system ensures accurate results and saves you time.

### Update a Book

Easily update your library with the "Update a Book" feature. Modify details like title, author, or edition to keep your collection current. It's a simple, user-friendly way to maintain accuracy.

### Delete a Book

The "Delete a Book" feature lets you remove books you no longer need. Just search for the book and delete it, helping you keep your library organized and relevant.

### Available Books

The "Available Books" feature gives you a clear overview of your entire collection, displaying all added



books with their details in one place for easy browsing.

## Significance and Benefits

The Book Tracker website offers multiple benefits for organizing and managing your collection:

1. **Enhanced Organization:** Keep your library tidy and easily accessible with detailed records.
2. **Time-Saving:** Quickly find and view books, eliminating the need for manual searches.
3. **Personalized Experience:** Customize your collection with specific details like author and edition.
4. **Flexibility:** Modify or remove books as your collection evolves.
5. **Scalability:** Can handle large collections, making it suitable for both casual readers and bibliophiles.
6. **Improved Reading Experience:** Track your reading progress and discover gaps in your library.
7. **Accessibility:** Manage your library from any device, anywhere.

In summary, the Book Tracker simplifies collection management and enhances the reading experience with its powerful features.

## User Management Module

**Purpose:** Manages user authentication, profiles, and secure access to personalized features.

- **Features:**
  - **User Registration:** Implements user sign-up with basic details (username, email, password) and optional profile personalization.
  - **Login/Logout:** Utilizes secure authentication mechanisms (e.g., JWT, OAuth) for login/logout functionality.
  - **Profile Management:** Allows users to update account details, preferences, and perform actions like password changes or account deletion.

### Benefits:

- **Secure Access:** Ensures secure user authentication and data protection using encryption and session management techniques.
- 
- **Personalization:** Adapts user experience by managing customizable preferences stored in the user profile.
- 
- **Accountability:** Tracks user actions, such as book additions, updates, and deletions, using unique user IDs.

## Schedule Management Module

**Purpose:** Facilitates the creation and tracking of user reading schedules and goals.

- **Features:**
  - **Add Reading Schedules:** Allows users to set start/end dates for books, integrating with a calendar or scheduling library (e.g., Cron jobs, Date-fns).

- **Track Progress:** Uses progress bars or percentage completion algorithms to display reading status.
- **Reminder Notifications:** Implements push notifications or email alerts via services like Firebase or Twilio.
- **Reading History:** Stores book start/end dates and ratings in a database (e.g., MongoDB) to track completed books.

#### Benefits:

- **Consistent Habits:** Uses time-based logic and notifications to prompt users towards completing reading goals.
- **Time Management:** Helps users balance reading schedules with other tasks via integration with external calendar APIs.
- **User Engagement:** Provides feedback on reading progress, boosting retention.

### Data Analytics & Reporting Module

**Purpose:** Provides data-driven insights into user reading behavior, library trends, and collection management.

- **Features:**
  - **Reading Analytics:** Utilizes data visualization libraries (e.g., Chart.js, D3.js) to display statistics like reading progress, time spent, and reading speed.
  - **Collection Insights:** Analyzes collection data (genres, authors, editions) using database queries and aggregation frameworks (e.g., MongoDB Aggregation Pipeline).
  - **Activity Reporting:** Tracks historical data on book additions, deletions, and updates in a time-series format.
  - **Customizable Reports:** Generates reports in PDF or CSV formats using server-side libraries (e.g., Puppeteer, csv-parser).

#### Benefits:

- **User Engagement:** Displays real-time analytics, encouraging continuous interaction with visual graphs and charts.
- **Self-Improvement:** Calculates metrics like reading speed and average time per book, helping users improve their habits.
- **Informed Decisions:** Provides insights into collection trends, helping users optimize their libraries by recommending genres, authors, or editions to focus on.

#### Additional Benefits:

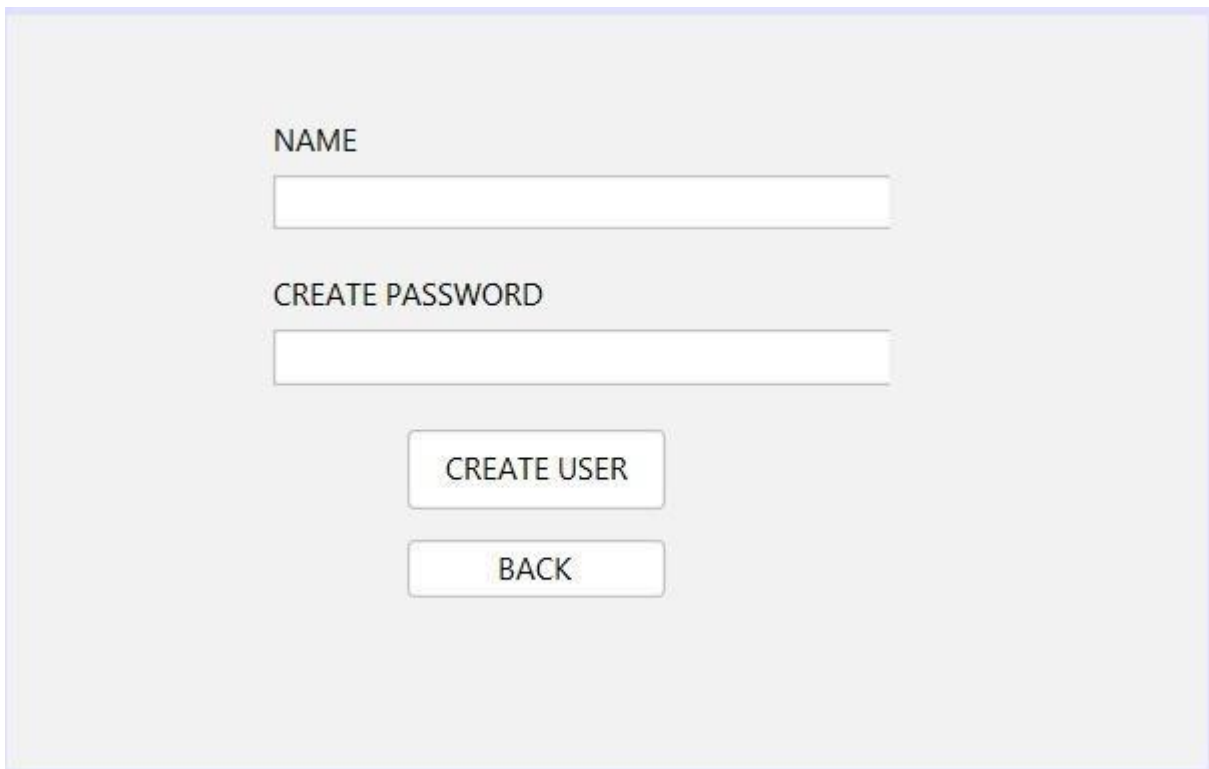
- **Motivation:** Visual goal tracking widgets and performance metrics provide real-time feedback on user progress.
- **Data Exporting:** Supports the export of insights into multiple formats, enhancing sharing capabilities and cross-platform usability.
- **Transparency:** Activity tracking provides a clear timeline of changes, aiding in maintaining collection integrity

#### Maintainability:

- 🔧 The system should be available 24/7 for administrators to perform tasks like adding, updating, or deleting student data and generating reports.
- 🔧 It should maintain high availability and perform critical operations, like querying student marks, with minimal downtime or service interruptions

## 1. DESIGN

### Login Register Page :



A user registration form with a light gray background and a thin blue border. It contains two text input fields, two buttons, and labels for each.

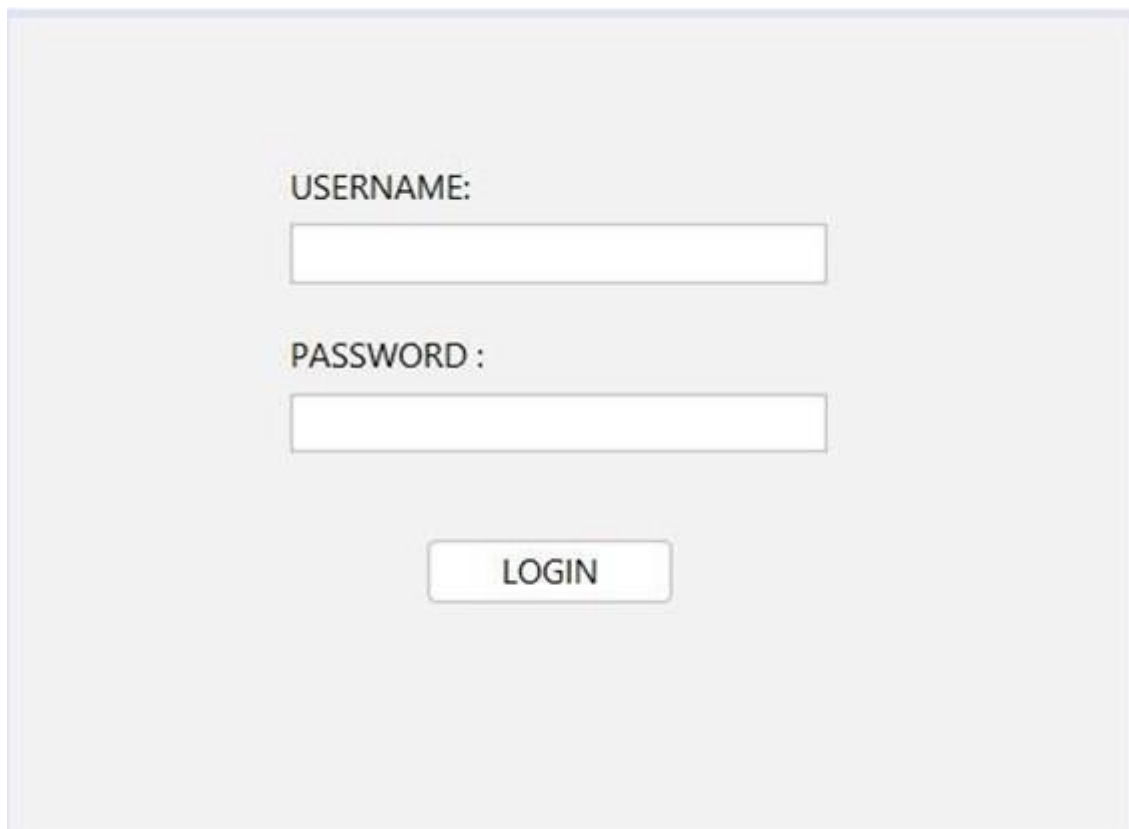
NAME

CREATE PASSWORD

CREATE USER

BACK

### User Logein Page:



A user login form with a light gray background and a thin blue border. It contains two text input fields, one button, and labels for each.

USERNAME:

PASSWORD :

LOGIN

### Add Book to Gallery Page :

ADD BOOK

NAME OF THE BOOK :

AUTHOR OF THE BOOK :

EDITION

ADD THE BOOK

BACK

### Find Your Book Page :

FIND THE BOOK

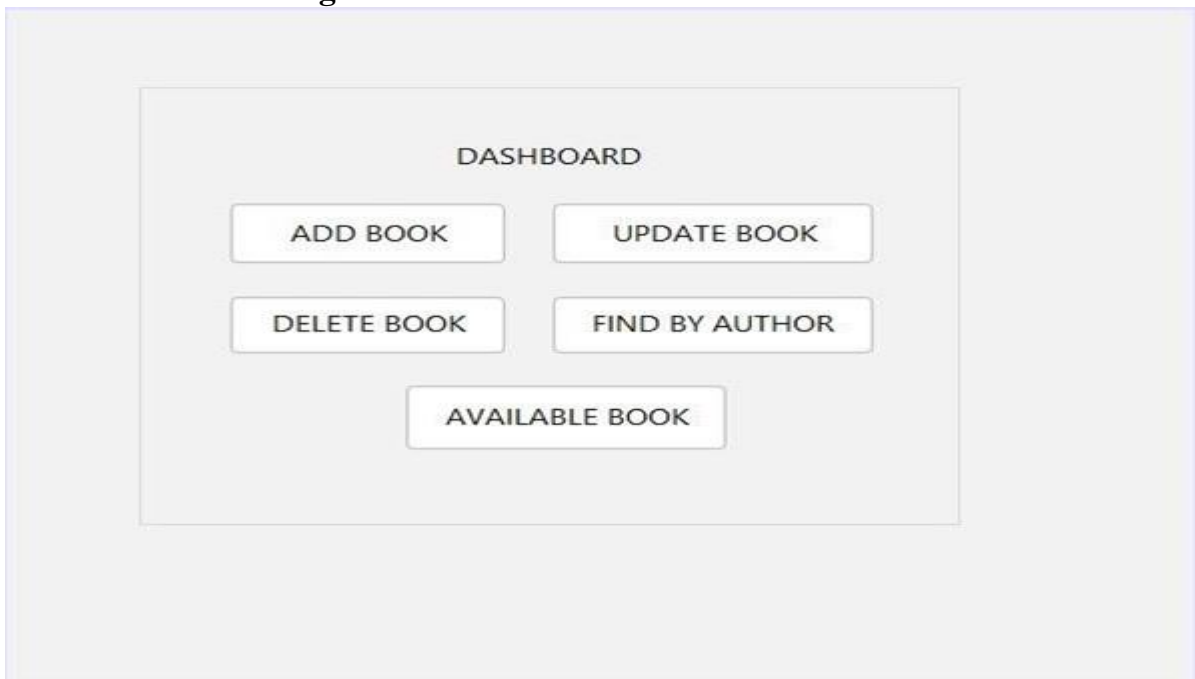
AUTHOR NAME :

FIND BOOK

BOOK	EDITION

BACK

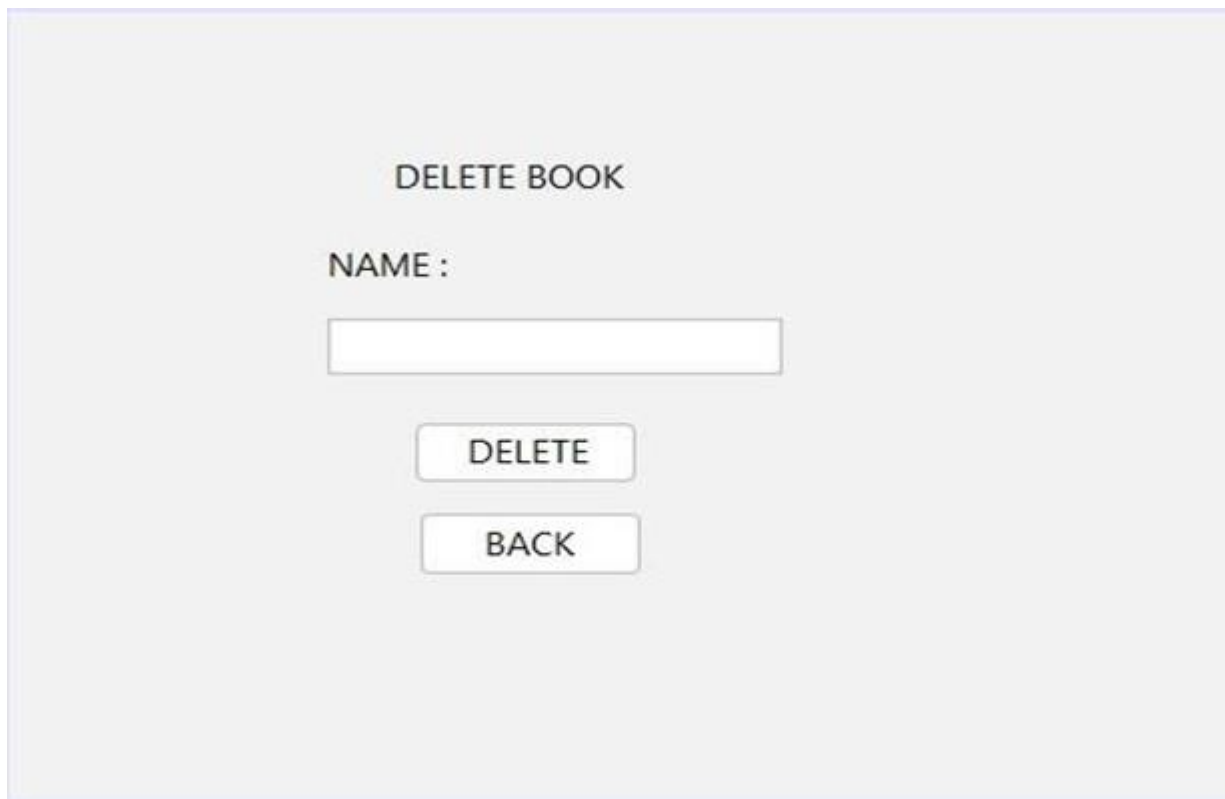
### Your DashBoard Page :



A screenshot of a web application dashboard. The dashboard is titled "DASHBOARD" and contains five buttons arranged in a grid. The buttons are: "ADD BOOK", "UPDATE BOOK", "DELETE BOOK", "FIND BY AUTHOR", and "AVAILABLE BOOK". The buttons are arranged in a 2x2 grid with "AVAILABLE BOOK" centered below the other four.

DASHBOARD	
ADD BOOK	UPDATE BOOK
DELETE BOOK	FIND BY AUTHOR
AVAILABLE BOOK	

### Delete Book Page :



A screenshot of a web application page titled "DELETE BOOK". The page contains a label "NAME :", a text input field, and two buttons: "DELETE" and "BACK".

DELETE BOOK	
NAME :	<input type="text"/>
DELETE	
BACK	

### Your Book List Page :

name	author	edition

### Update Your Book Page :

UPDATE THE BOOK

NAME OF THE BOOK :

AUTHOR :

EDITION :

## 4. IMPLEMENTATION

CODE :

```
import com.mongodb.MongoClientSettings; import
com.mongodb.client.MongoClients; import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import org.bson.Document;

/**
 *
 *      @author harsh
 */ public class train_adder extends javax.swing.JFrame {

/**
 *      Creates new form train_adder
 */ public train_adder() { 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() {
jLabel2 = new javax.swing.JLabel(); jLabel1 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField(); jLabel3 = new
javax.swing.JLabel(); jTextField2 = new javax.swing.JTextField(); jLabel4 =
new javax.swing.JLabel(); jTextField3 = new javax.swing.JTextField();
jLabel5 = new javax.swing.JLabel(); jTextField4 = new
javax.swing.JTextField(); jLabel6 = new javax.swing.JLabel(); jTextField5 =
new javax.swing.JTextField(); jLabel7 = new javax.swing.JLabel();
jTextField6 = new javax.swing.JTextField(); jLabel8 = new
javax.swing.JLabel(); jTextField7 = new javax.swing.JTextField(); jButton1 =
new javax.swing.JButton(); jButton2 = new javax.swing.JButton();
```

```
jLabel2.setText("jLabel2");  
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)  
; j
```

**Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license**

**\* Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template \*/**

```
package com.mycompany.sample;
```

```
import java.io.File;
```

```
/**
```

```
*
```

```
* @author jevan
```

```
*/
```

```
public class FileModel {
```

```
    private static FileModel instance;
```

```
    private File selectedFile;
```

```
    // Private constructor to enforce singleton pattern    private FileModel() {}
```

```
    public static FileModel getInstance() {        if (instance == null) {  
        instance = new FileModel();  
    }  
}
```

```
    return instance;
```

```
}
```

```
public File getSelectedFile() {
```

```
    return selectedFile;
```



```

    }

    public void setSelectedFile(File selectedFile) {        this.selectedFile =
selectedFile;

    }

}
/*
 *      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 com.mycompany.sample;

/**
 *
 *      @author jevan
 */
import com.mongodb.MongoClientSettings; import
com.mongodb.MongoClientURI; import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import org.bson.Document; import
javax.swing.JOptionPane; import java.util.Arrays;

public class NewJFrame extends javax.swing.JFrame {

    /**
     *      Creates new form NewJFrame
     */
    public NewJFrame() {

        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() {  
  
    jPanel1 = new javax.swing.JPanel();    jLabel1 = new  
javax.swing.JLabel();    jLabel3 = new javax.swing.JLabel();  
jTextField1 = new javax.swing.JTextField();    jLabel2 = new  
javax.swing.JLabel();    jButton1 = new javax.swing.JButton();  
jTextField2 = new javax.swing.JTextField();    jButton2 = new  
javax.swing.JButton();  
  
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)  
;  
  
    jLabel1.setText("NAME");  
  
    jTextField1.addActionListener(new java.awt.event.ActionListener() {  
public void actionPerformed(java.awt.event.ActionEvent evt) {  
jTextField1ActionPerformed(evt);  
    }  
});  
  
    jLabel2.setText("CREATE PASSWORD");  
  
    jButton1.setText("CREATE USER");  
    jButton1.addActionListener(new java.awt.event.ActionListener() {  
public void actionPerformed(java.awt.event.ActionEvent evt) {  
jButton1ActionPerformed(evt);  
    }  
});  
  
    jButton2.setText("BACK");  
    jButton2.addActionListener(new java.awt.event.ActionListener() {  
public void actionPerformed(java.awt.event.ActionEvent evt) {  
jButton2ActionPerformed(evt);
```

$$\left. \begin{array}{l} \} \\ \} \end{array} \right);$$

```

        javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);        jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LE
ADING)        .addGroup(jPanel1Layout.createSequentialGroup()
                .addContainerGap()
                .addGroup(jPanel1Layout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING)
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jTextField1)
                        .addGap(182, 182, 182)
                        .addComponent(jLabel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
176, javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(jPanel1Layout.createSequentialGroup()
                        .addComponent(jTextField2)
                        .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addGap(0, 0, Short.MAX_VALUE))
                    .addComponent(jLabel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addGap(60, 60, 60)
                    .addGroup(jPanel1Layout.createParallelGroup(javax.swing.Gro
upLayout.Alignment.LEADING, false)
                        .addComponent(jButton1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                        .addComponent(jButton2,
javax.swing.GroupLayout.DEFAULT_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
    .addGap(0, 0, Short.MAX_VALUE))
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jButton1,
javax.swing.GroupLayout.PREFERRED_SIZE, 32,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jButton2)
    .addContainerGap(10, 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()
        .addGap(99, 99, 99)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 253,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(126, Short.MAX_VALUE))
    );
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(35, 35, 35)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(58, Short.MAX_VALUE))
    );

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

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    String uri = "Connection string for mongodb";

    // Create a new MongoClient using the connection string    try
    (MongoClient mongoClient = MongoClient.create(uri)) {

        // Connect to the "PRACTICE" database
        MongoDB database = mongoClient.getDatabase("PRACTICE");

        // Get the "example" collection
        MongoCollection<Document> collection =
database.getCollection("example");

```

```

        // Create documents to insert
        Document doc1 = new Document("name", jTextField1.getText())
            .append("password",jTextField2.getText());

        // Insert multiple documents at once        collection.insertOne(doc1);

        System.out.println("Documents inserted successfully");
        JOptionPane.showMessageDialog(this,"created user successfully");

    }
}

private void jTextField1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    add_book x = new add_book();        x.setVisible(true);
    this.dispose();
}

/**
 * @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(NewJFrame.class.getName()).log(java.util.l
ogging.Level.
SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.l
ogging.Level.
SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.l
ogging.Level.
SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(NewJFrame.class.getName()).log(java.util.l
ogging.Level. SEVERE, null, ex);
    }
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {      public void
run() {
        new NewJFrame().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.JLabel jLabel3;    private javax.swing.JPanel jPanel1;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField2;
    // End of variables declaration
}

```

```

/*
 *      Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 */

package com.mycompany.sample;

/**
 *
 *      @author jevan
 */
public class Sample {

    public static void main(String[] args) {        add_book a = new add_book();
a.setVisible(true);
    }
}

/*
 *      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 com.mycompany.sample;

/**
 *
 *      @author jevan
 */
import com.mongodb.client.MongoClient; import
com.mongodb.client.MongoClients; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import javax.swing.JOptionPane;
import org.bson.Document;

```



```

public class add_book extends javax.swing.JFrame {

    /**
     * Creates new form add_book
     */
    public add_book() {
        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() {

        jLabel3 = new javax.swing.JLabel();        jButton2 = new
javax.swing.JButton();        jButton3 = new javax.swing.JButton();
jPanel1 = new javax.swing.JPanel();        jButton1 = new
javax.swing.JButton();        jLabel1 = new javax.swing.JLabel();
jTextField1 = new javax.swing.JTextField();        jTextField2 = new
javax.swing.JTextField();        jLabel2 = new javax.swing.JLabel();

        jLabel3.setText("jLabel3");

        jButton2.setText("jButton2");

        jButton3.setText("jButton3");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        ;

        jButton1.setText("LOGIN");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jButton1ActionPerformed(evt);
            }
        });
    }
}

```

```

    }
});

jLabel1.setText("USERNAME:");

jLabel2.setText("PASSWORD : ");

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);    jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel1,
                javax.swing.GroupLayout.PREFERRED_SIZE, 98,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField1)
            .addComponent(jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE, 76,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField2,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                191, javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addGap(49, 49, 49)
                .addComponent(jButton1,
                    javax.swing.GroupLayout.PREFERRED_SIZE, 87,
                    javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(191, false)
    )
    .addGap(10, 10, 10)
);
jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jLabel1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField1)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jLabel2)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jTextField2)
                .addGap(10, 10, 10)
            )
            .addGroup(jPanel1Layout.createSequentialGroup()
                .addComponent(jButton1)
                .addGap(10, 10, 10)
            )
        )
    )
);

```

```

        .addContainerGap()
        .addComponent(jLabel1)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        )
        .addComponent(jTextField1,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)
        .addComponent(jLabel2)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
        )
        .addComponent(jTextField2,
        javax.swing.GroupLayout.PREFERRED_SIZE,
        javax.swing.GroupLayout.DEFAULT_SIZE,
        javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(32, 32, 32)
        .addComponent(jButton1)
        .addContainerGap()
    );

    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(93, 93, 93)
            .addComponent(jPanel1,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(104, Short.MAX_VALUE))
        );
    layout.setVerticalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
            .addGap(47, 47, 47)
            .addComponent(jPanel1,
                javax.swing.GroupLayout.PREFERRED_SIZE,

```

```

javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addContainerGap(80, Short.MAX_VALUE))
    );

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

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    String uri = "Connecting string mongodb"; // Replace with your actual
MongoDB URI

    // MongoDB connection and operation
    try (MongoClient mongoClient = MongoClient.create(uri)) {
        // Connect to the "PRACTICE" database
        MongoDB database = mongoClient.getDatabase("PRACTICE");

        // Get the "users" collection
        MongoCollection<Document> collection =
database.getCollection("example");

        // Query the database for a document with the matching username and
password
        Document query = new Document("name",
jTextField1.getText()).append("password", jTextField2.getText());
        Document user = collection.find(query).first();
        Document admin = new Document("name",
"admin").append("password", "admin");

        if (admin.equals(query)){
            JFrame x = new JFrame();
            x.setVisible(true);
this.dispose();
        }
        else if (user != null) {
            // User found, credentials are correct
            JFrame dash = new JFrame();
dash.setVisible(true);
this.dispose();
        } else {
            // User not found, credentials are incorrect

```

```

        JOptionPane.showMessageDialog(this,"give the correct password");
    }
}

/**
 * @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(add_book.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(add_book.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(add_book.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(add_book.class.getName()).log(java.util.log
ging.Level.SEVERE, null, ex);

```

```

    }
    //</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {      public void
run() {
        new add_book().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.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;  private javax.swing.JPanel jPanel1;
private javax.swing.JTextField jTextField1;  private javax.swing.JTextField
jTextField2;
    // End of variables declaration
}

/*
 *      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 com.mycompany.sample;

import com.mongodb.client.MongoClient; import
com.mongodb.client.MongoClients; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import javax.swing.JOptionPane;
import org.bson.Document;

/**

```

```

*
*      @author jevan
*/
public class addbook extends javax.swing.JFrame {

    /**
    *      Creates new form addbook
    */
    public addbook() {        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() {

        jLabel4 = new javax.swing.JLabel();
        buttonGroup1 = new javax.swing.ButtonGroup();        jScrollBar1 = new
javax.swing.JScrollBar();        jPanel1 = new javax.swing.JPanel();
        jButton1 = new javax.swing.JButton();        jLabel2 = new
javax.swing.JLabel();        jTextField3 = new javax.swing.JTextField();
        jLabel5 = new javax.swing.JLabel();        jTextField1 = new
javax.swing.JTextField();        jLabel1 = new javax.swing.JLabel();
        jTextField2 = new javax.swing.JTextField();        jLabel3 = new
javax.swing.JLabel();        jButton2 = new javax.swing.JButton();

        jLabel4.setText("jLabel4");

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
        ;

        jButton1.setText("ADD THE BOOK");
        jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {

```

```

jButton1ActionPerformed(evt);
    }
});

jLabel2.setText("NAME OF THE BOOK :");

jLabel5.setText("EDITION");

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("ADD BOOK");

jLabel3.setText("AUTHOR OF THE BOOK :");

javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);    jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(10, 10, 10)
        .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jLabel2,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                170, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField1)
            .addComponent(jLabel3,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                159, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jTextField2)

```



```

        .addComponent(jLabel5,
javax.swing.GroupLayout.PREFERRED_SIZE,
135, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE, 247,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(73, 73, 73)
        .addComponent(jButton1))))
        .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(49, 49, 49)
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 159,
javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addContainerGap()
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

        .addGroup(jPanel1Layout.createSequentialGroup()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(12, 12, 12)
        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel3)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jTextField2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
    .addComponent(jLabel5)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
    .addComponent(jTextField3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
    .addComponent(jButton1)
    .addContainerGap()
);

    jButton2.setText("BACK");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
}
});

    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(65, 65, 65)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

```

```

        .addGroup(layout.createSequentialGroup()
            .addGap(145, 145, 145)
            .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE,
111, javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap(76, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
    .addGap(22, 22, 22)
    .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

    .addComponent(jButton2)
    .addContainerGap(13, Short.MAX_VALUE))
    );

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

```

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    // TODO add your handling code here:
    String uri = "connecting string mongodb";

    // Create a new MongoClient using the connection string      try
    (MongoClient mongoClient = MongoClient.create(uri)) {

        // Connect to the "PRACTICE" database
        MongoDB database = mongoClient.getDatabase("PRACTICE");

        // Get the "example" collection
        MongoCollection<Document> collection =

```

```

database.getCollection("BOOK");

    // Create documents to insert
    Document doc1 = new Document("name",
jTextField1.getText().toLowerCase())
        .append("author",jTextField2.getText().toLowerCase())
        .append("edition", jTextField3.getText().toLowerCase());

    // Insert multiple documents at once        collection.insertOne(doc1);
jTextField1.setText(null);        jTextField2.setText(null);
jTextField3.setText(null);
    System.out.println("Documents inserted successfully");
    JOptionPane.showMessageDialog(this,"inserted the book
successfully");

    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    this.dispose();
}

/**
 * @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(addbook.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(addbook.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(addbook.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

    } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(addbook.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 addbook().setVisible(true);
    }
});
}

    // Variables declaration - do not modify      private
javax.swing.ButtonGroup buttonGroup1;    private javax.swing.JButton
jButton1;    private javax.swing.JButton jButton2;    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.JPanel jPanel1;    private
javax.swing.JScrollBar jScrollPane1;    private javax.swing.JTextField
jTextField1;    private javax.swing.JTextField jTextField2;    private
javax.swing.JTextField jTextField3;
    // End of variables declaration
}

```

```

/*
 *      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 com.mycompany.sample;

/**
 *
 *      @author jevan
 */
import com.mongodb.client.MongoClient; import
com.mongodb.client.MongoClients; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoCursor; import
com.mongodb.client.MongoDatabase; import org.bson.Document;

import javax.swing.*;
import javax.swing.table.DefaultTableModel; public class authorfind extends
javax.swing.JFrame {

    /**
 *      Creates new form authorfind
 */
    public authorfind() {      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();    JLabel2 = new
javax.swing.JLabel();    JTextField1 = new javax.swing.JTextField();
JButton1 = new javax.swing.JButton();    JScrollPane1 = new
javax.swing.JScrollPane();    jTable1 = new javax.swing.JTable();
    JButton2 = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
;

    JLabel1.setText("FIND THE BOOK");

    JLabel2.setText("AUTHOR NAME :");

    JButton1.setText("FIND BOOK");
    JButton1.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
JButton1ActionPerformed(evt);
    }
});

    jTable1.setModel(new javax.swing.table.DefaultTableModel(        new
Object [][] {            {null, null},
                {null, null},
                {null, null}
            },
            new String [] {
                "BOOK", "EDITION"
            }
        ));
    JScrollPane1.setViewportViewView(jTable1);

    JButton2.setText("BACK");
    JButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
JButton2ActionPerformed(evt);

```

```

    }
});

    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()
            .addComponent(jLabel1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 96,
    javax.swing.GroupLayout.PREFERRED_SIZE))
            .addGroup(layout.createSequentialGroup()
                .addGap(22, 22, 22)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
    LEADING)
        .addComponent(jLabel2,
    javax.swing.GroupLayout.PREFERRED_SIZE,
    135, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGroup(layout.createSequentialGroup()
                .addComponent(jTextField1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 178,
    javax.swing.GroupLayout.PREFERRED_SIZE)
                .addGap(18, 18, 18)
                .addComponent(jButton1))
            .addComponent(jScrollPane1,
    javax.swing.GroupLayout.PREFERRED_SIZE, 339,
    javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addGroup(layout.createSequentialGroup()
                .addGap(136, 136, 136)
                .addComponent(jButton2)))
        .addContainerGap(31, Short.MAX_VALUE))
    );

```



```

        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 26,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
BASELINE)
        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(jButton1))
        .addGap(31, 31, 31)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 151,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
        .addComponent(jButton2)
        .addContainerGap(8, Short.MAX_VALUE))
    );

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

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    String URI = "Connecting string mongodb";
    DefaultTableModel model = (DefaultTableModel) jTable1.getModel();
    model.setRowCount(0); // Clear existing data    String authorName =

```

```

jTextField1.getText();
    try (MongoClient mongoClient = MongoClient.create(URI)) {
        MongoDB database = mongoClient.getDatabase("PRACTICE");
        MongoCollection<Document> collection =
database.getCollection("BOOK");

        // Define the filter based on author name
        Document filter = new Document("author", authorName);

        try (MongoCursor<Document> cursor =
collection.find(filter).iterator()) {            boolean found = false;
while (cursor.hasNext()) {                found = true;
            Document doc = cursor.next();
            String name = doc.getString("name");
            String edition = doc.getString("edition");

            model.addRow(new Object[]{name, edition});
        }

        if (!found) {
            JOptionPane.showMessageDialog(this, "No books found for the
author: " + authorName);
        } else {
            JOptionPane.showMessageDialog(this, "Books by " + authorName
+ " loaded successfully!");
        }
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    this.dispose();
}

/**
 * @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(authorfind.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(authorfind.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(authorfind.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(authorfind.class.getName()).log(java.util.lo
gging.Level.SEVERE, null, ex);
        }
    }
}
//</editor-fold>

    /* Create and display the form */
    java.awt.EventQueue.invokeLater(new Runnable() {          public void
run() {
        new authorfind().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; private javax.swing.JTextField jTextField1;
    // End of variables declaration
}

/*
 * 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 com.mycompany.sample;

/**
 *
 * @author jevan
 */
public class dash extends javax.swing.JFrame {

    /**
     * Creates new form dash
     */
    public dash() {
        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() {

    jPanel1 = new javax.swing.JPanel();    jPanel3 = new
javax.swing.JPanel();    jPanel2 = new javax.swing.JPanel();    jButton2
= new javax.swing.JButton();    jButton4 = new javax.swing.JButton();
jButton6 = new javax.swing.JButton();    jButton3 = new
javax.swing.JButton();    jButton5 = new javax.swing.JButton();
jLabel1 = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
;

    jButton2.setText("ADD BOOK");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
    }
});

    jButton4.setText("DELETE BOOK");
    jButton4.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton4ActionPerformed(evt);
    }
});

    jButton6.setText("AVAILABLE BOOK");
    jButton6.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton6ActionPerformed(evt);
    }
});

    jButton3.setText("UPDATE BOOK");
    jButton3.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton3ActionPerformed(evt);
    }
}
}

```

```

});

jButton5.setText("FIND BY AUTHOR");
jButton5.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton5ActionPerformed(evt);
}
});

    javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);    jPanel2.setLayout(jPanel2Layout);
jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()
        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jButton4,
                javax.swing.GroupLayout.DEFAULT_SIZE,
                javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jButton2,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                107, javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)

        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(jButton3,
                javax.swing.GroupLayout.PREFERRED_SIZE,
                124, javax.swing.GroupLayout.PREFERRED_SIZE)
            .addComponent(jButton5)))
        .addGroup(jPanel2Layout.createSequentialGroup()
            .addGap(68, 68, 68)

```

```

        .addComponent(jButton6)))
    .addContainerGap()
);
jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel2Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jButton2,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton3,
javax.swing.GroupLayout.PREFERRED_SIZE, 33,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
    .addComponent(jButton4,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jButton5,
javax.swing.GroupLayout.PREFERRED_SIZE, 31,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jButton6,
javax.swing.GroupLayout.PREFERRED_SIZE, 35,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap()
);

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);
jLabel1.setText("DASHBOARD");

javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);    jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(

```

```

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGap(85, 85, 85)
            .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addContainerGap())
    );
jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel3Layout.createSequentialGroup()
        .addContainerGap()
        .addComponent(jLabel1,
javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
    .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap())
    );

    javax.swing.GroupLayout jPanel1Layout = new
    javax.swing.GroupLayout(jPanel1);    jPanel1.setLayout(jPanel1Layout);
    jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()

```



```

        .addGap(23, 23, 23)
        .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(22, Short.MAX_VALUE))
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(jPanel1Layout.createSequentialGroup()
        .addGap(18, 18, 18)
        .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(30, 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()
        .addGap(50, 50, 50)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(89, Short.MAX_VALUE))
    );
    layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addGap(42, 42, 42)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,

```

```

javax.swing.GroupLayout.PREFERRED_SIZE)
        .addContainerGap(86, Short.MAX_VALUE))
    );

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

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    addbook x = new addbook();    x.setVisible(true);
}
private void
jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:    updatedelete q = new
updatedelete();    q.setVisible(true);
}

private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    delete v = new delete();    v.setVisible(true);
}

private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    authorfind w = new authorfind();    w.setVisible(true);
}

private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    showbook r = new showbook();    r.setVisible(true);
}

/**
 * @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(dash.class.getName()).log(java.util.logging.  
    Level.SEVERE, null, ex);  
    } catch (InstantiationException ex) {  
    java.util.logging.Logger.getLogger(dash.class.getName()).log(java.util.logging.  
    Level.SEVERE, null, ex);  
    } catch (IllegalAccessException ex) {
```

```
    java.util.logging.Logger.getLogger(dash.class.getName()).log(java.util.logging.  
    Level.SEVERE, null, ex);  
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
    java.util.logging.Logger.getLogger(dash.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 dash().setVisible(true);  
    }  
    });  
    }
```

```
    // Variables declaration - do not modify private  
    javax.swing.JButton jButton2; private javax.swing.JButton jButton3;  
    private javax.swing.JButton jButton4; private javax.swing.JButton  
    jButton5; private javax.swing.JButton jButton6; private
```

```

javax.swing.JLabel jLabel1;    private javax.swing.JPanel jPanel1;    private
javax.swing.JPanel jPanel2;    private javax.swing.JPanel jPanel3;
    // End of variables declaration
}


/*
 *      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 com.mycompany.sample;


/**
 *
 *      @author jevan
*/

import com.mongodb.client.MongoClient; import
com.mongodb.client.MongoClients; import
com.mongodb.client.MongoCollection; import
com.mongodb.client.MongoDatabase; import org.bson.Document;


import javax.swing.*;
public class delete extends javax.swing.JFrame {


    /**
 *      Creates new form delete
 */
    public delete() {        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() {

    jPanel1 = new javax.swing.JPanel();    jLabel1 = new
javax.swing.JLabel();    jButton1 = new javax.swing.JButton();    jLabel2
= new javax.swing.JLabel();    jTextField1 = new javax.swing.JTextField();
jButton2 = new javax.swing.JButton();

    setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE)
    ;

    jLabel1.setText("NAME :");

    jButton1.setText("DELETE");
    jButton1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButton1ActionPerformed(evt);
        }
    });

    jLabel2.setText("DELETE BOOK");

    javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);    jPanel1.setLayout(jPanel1Layout);
jPanel1Layout.setHorizontalGroup(

    jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .add(jLabel1)
                    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                    .add(jButton1)
                    .add(jLabel2)
                    .add(jTextField1)
                    .add(jButton2)
                    .addContainerGap(10, true))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .add(jLabel2)
                    .addContainerGap(10, true))
            ), javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .add(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .add(jLabel1)
                        .add(jButton1)
                        .add(jLabel2)
                        .add(jTextField1)
                        .add(jButton2)
                    )
                    .addContainerGap(10, true))
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .add(jLabel2)
                    .addContainerGap(10, true))
            ), javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
    );
    jPanel1Layout.setVerticalGroup(
        javax.swing.GroupLayout(jPanel1).getLayout()
    );
    jPanel1Layout.setSize(new Dimension(400, 300));
    jPanel1.setVisible(true);
}


```

```

        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE, 149,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(21, 21, 21)
            .addComponent(jLabel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
91, javax.swing.GroupLayout.PREFERRED_SIZE))))
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addGap(35, 35, 35)
            .addComponent(jButton1)))
        .addContainerGap()
    );
    jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel1Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabel2)
            .addGap(18, 18, 18)
            .addComponent(jLabel1)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
        .addComponent(jTextField1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jButton1)
            .addContainerGap()
        );

    jButton2.setText("BACK");
    jButton2.addActionListener(new java.awt.event.ActionListener() {
public void actionPerformed(java.awt.event.ActionEvent evt) {
jButton2ActionPerformed(evt);
        }
    });

```

```

        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(97, 97, 97)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
        .addGap(133, 133, 133)
        .addComponent(jButton2)))
        .addContainerGap(142, Short.MAX_VALUE))
);
layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
.addGroup(layout.createSequentialGroup()
        .addGap(48, 48, 48)
        .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(jButton2)
        .addContainerGap(86, Short.MAX_VALUE))
);

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

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:

```

```

    String URI = "connection string mongodb";
    String bookName = jTextField1.getText().trim();

    if (bookName.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Please enter a book name to delete.");
    }

    try (MongoClient mongoClient = MongoClient.create(URI)) {
        MongoDB database = mongoClient.getDatabase("PRACTICE");
        MongoCollection<Document> collection =
            database.getCollection("BOOK");

        // Define filter based on book name
        Document filter = new Document("name", bookName);

        // Check if the book exists
        Document book = collection.find(filter).first();    if (book == null) {
            JOptionPane.showMessageDialog(this, "Book not found.");
        }

        // If book exists, proceed with deletion    var deleteResult =
        collection.deleteOne(filter);    if (deleteResult.getDeletedCount() > 0) {
            JOptionPane.showMessageDialog(this, "Book deleted successfully!");
        } else {
            JOptionPane.showMessageDialog(this, "Failed to delete the book.");
        }
    }
}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
// TODO add your handling code here:
    this.dispose();
}

/**
 *    @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(delete.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(delete.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(delete.class.getName()).log(java.util.loggin
g.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(delete.class.getName()).log(java.util.loggin
g.Level.SEVERE
RE, null, ex);
        }
    }
}
//</editor-fold>

/* Create and display the form */
java.awt.EventQueue.invokeLater(new Runnable() {      public void
run() {
        new delete().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.JPanel jPanel1; private javax.swing.JTextField
jTextField1;
// End of variables declaration
}

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to
edit this template
 */
package com.mycompany.sample;

import java.awt.Color; import java.awt.Component; import java.awt.Font;
import java.awt.Image;
import java.awt.image.BufferedImage; import javax.swing.BorderFactory;
import javax.swing.Box; import javax.swing.BoxLayout; import
javax.swing.ImageIcon; import javax.swing.JButton; import
javax.swing.JFrame; import javax.swing.JLabel;
import javax.swing.JPanel;

/**
 *
 * @author jevan
 */
public class details {
    public static void showDetailView(String title, String description,
BufferedImage image) {
        JFrame detailFrame = new JFrame("Detail View");
        detailFrame.setSize(400, 400);
        detailFrame.setLocationRelativeTo(null);
    }
}

```

```

    JPanel detailPanel = new JPanel();
    detailPanel.setLayout(new BoxLayout(detailPanel, BoxLayout.Y_AXIS));
    detailPanel.setBackground(Color.WHITE);
    detailPanel.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));

    JLabel titleLabel = new JLabel(title);
    titleLabel.setFont(new Font("Arial", Font.BOLD, 16));
    titleLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    detailPanel.add(titleLabel);

    ImageIcon imageIcon = new ImageIcon(image.getScaledInstance(200, 150,
    Image.SCALE_SMOOTH));
    JLabel imageLabel = new JLabel(imageIcon);
    imageLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    detailPanel.add(imageLabel);

    JLabel descriptionLabel = new JLabel("<html><p style='width:300px;'>" +
    description +
    "</p></html>");
    descriptionLabel.setFont(new Font("Arial", Font.PLAIN, 14));
    descriptionLabel.setAlignmentX(Component.CENTER_ALIGNMENT);
    detailPanel.add(descriptionLabel);

    JButton closeButton = new JButton("Close");
    closeButton.setAlignmentX(Component.CENTER_ALIGNMENT);
    closeButton.addActionListener(e -> detailFrame.dispose());
    detailPanel.add(Box.createVerticalStrut(10));
    detailPanel.add(closeButton);

    detailFrame.add(detailPanel);    detailFrame.setVisible(true);
}
}

```

## **Conclusion:**

The “Online Book Tracking System” project, developed with attention to detail, the design emphasizes a seamless user experience while ensuring efficient functionality for administrators. Key features include adding, viewing, and managing users book, with the ability to perform statistical analysis and generate reports. The system maintains robust security, especially in sensitive operations like modifying and removing administrator accounts, to ensure data integrity and safeguard against unauthorized access. This system is designed to meet the current needs of educational perpose while being adaptable for future improvements, ensuring long-term effectiveness and flexibility in handling student performance data.

## **Reference links:**

- [1] <https://www.javatpoint.com/java-awt>  
<https://www.javatpoint.com/java-sw>