NEWS PORTAL SYSTEM

A PROJECT REPORT

Submitted by

YOGA S (2303811710422186)

in partial fulfillment of requirements for the award of the course

CGB1201 - JAVA PROGRAMMING

In

COMPUTER SCIENCE AND ENGINEERING

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

NOVEMBER- 2024

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **" NEWS PORTAL SYSTEM"** is the bonafide work of **YOGA S (2303811710422186)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE	**SIGNATURE**
Dr.A.Delphin Carolina Rani, M.E.,Ph.D.,	Mr. A. Malarmannan, M.E.,
HEAD OF THE DEPARTMENT	**SUPERVISOR**
PROFESSOR	ASSISTANT PROFESSOR
Department of CSE	Department of CSE
K.Ramakrishnan College of Technology (Autonomous)	K.Ramakrishnan College of Technology (Autonomous)
Samayapuram–621112.	Samayapuram–621112.

Submitted for the viva-voce examination held on 06/12/24

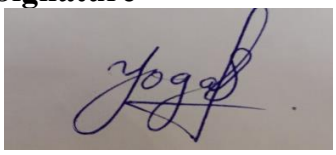
INTERNAL EXAMINER EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"NEWS PORTAL SYSTEM"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **CGB1201 - JAVA PROGRAMMING.**

.

Signature



YOGA S

Place: Samayapuram

Date:06/12/24

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. A. DELPHIN CAROLINA RANI, M.E.,Ph.D.,** Head of the department, **COMPUTER SCIENCE AND ENGINEERING** for providing her encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **MR. A. MALARMANNAN, M.E.,** Department of **COMPUTER SCIENCE AND ENGINEERING,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

VISION OF THE INSTITUTION

To serve the society by offering top-notch technical education on par with global standards

MISSION OF THE INSTITUTION

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of the industry and society.
- Be an institute with world class research facilities
- Be an institute nurturing talent and enhancing the competency of students to transform them as all-round personality respecting moral and ethical values

VISION OF DEPARTMENT

To be a center of eminence in creating competent software professionals with research and innovative skills.

MISSION OF DEPARTMENT

M1: Industry Specific: To nurture students in working with various hardware and software platforms inclined with the best practices of industry.

M2: Research: To prepare students for research-oriented activities.

M3: Society: To empower students with the required skills to solve complex technological problems of society.

PROGRAM EDUCATIONAL OBJECTIVES

1. PEO1: Domain Knowledge

To produce graduates who have strong foundation of knowledge and skills in the field of Computer Science and Engineering.

2. PEO2: Employability Skills and Research

To produce graduates who are employable in industries/public sector/research organizations or work as an entrepreneur.

3. PEO3: Ethics and Values

To develop leadership skills and ethically collaborate with society to tackle real-world challenges.

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO 1: Domain Knowledge

To analyze, design and develop computing solutions by applying foundational concepts of Computer Science and Engineering.

PSO 2: Quality Software

To apply software engineering principles and practices for developing quality software for scientific and business applications.

PSO 3: Innovation Ideas

To adapt to emerging Information and Communication Technologies (ICT) to innovate ideas and solutions to existing/novel problems

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

This Java code implements a News Portal application using a graphical user interface (GUI) with the AWT library. The application consists of a login, registration, user dashboard, and admin panel, each designed using a CardLayout to allow for seamless switching between screens. Users can register with a chosen role (either User or Admin), and after logging in, they are redirected to the appropriate dashboard based on their role. The User dashboard allows access to news articles categorized by topics like Politics and Sports. Admin users have additional privileges to manage articles, including adding or deleting them. The application uses HashMap to store user credentials, roles, and articles. It also includes input validation and confirmation dialogs to enhance user experience. Overall, the portal serves as a simple news management system with a role-based access control mechanism.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD JAVA APPLICATIONS FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
This Java code implements a News Portal application using a graphical user interface (GUI) with the AWT library. The application consists of a login, registration, user dashboard, and admin panel, each designed using a `CardLayout` to allow for seamless switching between screens. Users can register with a chosen role (either User or Admin), and after logging in, they are redirected to the appropriate dashboard based on their role. The User dashboard allows access to news articles categorized by topics like Politics and Sports. Admin users have additional privileges to manage articles, including adding or deleting them. The application uses `HashMap` to store user credentials, roles, and articles. It also includes input validation and confirmation dialogs to enhance user experience. Overall, the portal serves as a simple news management system with a role-based access control mechanism.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Objective

The objective of the provided code is to create a simple role-based News Portal application using Java's AWT framework. The application allows users to log in or register as either "Admin" or "User," with distinct functionalities for each role. Admins can manage articles by adding, viewing, or deleting them across predefined categories, while Users can browse and read articles. The system utilizes an intuitive Card Layout to navigate between login, registration, dashboard, and admin panels. User credentials, roles, and articles are stored in memory using HashMap, ensuring straightforward data handling. The interface incorporates interactive elements like buttons, lists, and dialogs to improve usability. This application demonstrates fundamental principles of role-based access control and content management in a graphical desktop environment.

1.2 Overview

The provided code implements a News Portal application in Java using the AWT framework, designed to manage articles and user roles. It supports two types of users: Admins and regular Users. Admins can add, view, and delete articles across various categories, while Users can browse and read articles. The application employs a Card Layout for seamless navigation between login, registration, dashboard, and admin functionalities. User data (credentials and roles) and articles are managed in memory using HashMap. The design leverages inner classes for data encapsulation, and interactive UI elements like lists, buttons, and dialogs enhance user experience. Overall, the application demonstrates a structured approach to role-based content management.

1.3 Java Programming Concepts

The provided code incorporates several key Java programming concepts:

1. AWT (Abstract Window Toolkit):

AWT is used for creating the graphical user interface (GUI) components like Frame,

Panel, Button, Label, TextField, Choice, TextArea, and List.

The layout manager CardLayout is used for switching between different views (Login, Registration, Dashboard, and Admin Panel).

2. Event Handling:

Event listeners, specifically ActionListener, are used to handle user actions (e.g., clicking buttons).

ItemListener is used for handling item selections in Choice components and list selections.

Collections Framework:

3. HashMap: Used for storing users (with their passwords), roles, and articles. The keys are String types, and values are appropriate data types like String (password, role) or ArrayList<Article> (list of articles).

ArrayList: Used to manage dynamic lists of articles for each category.

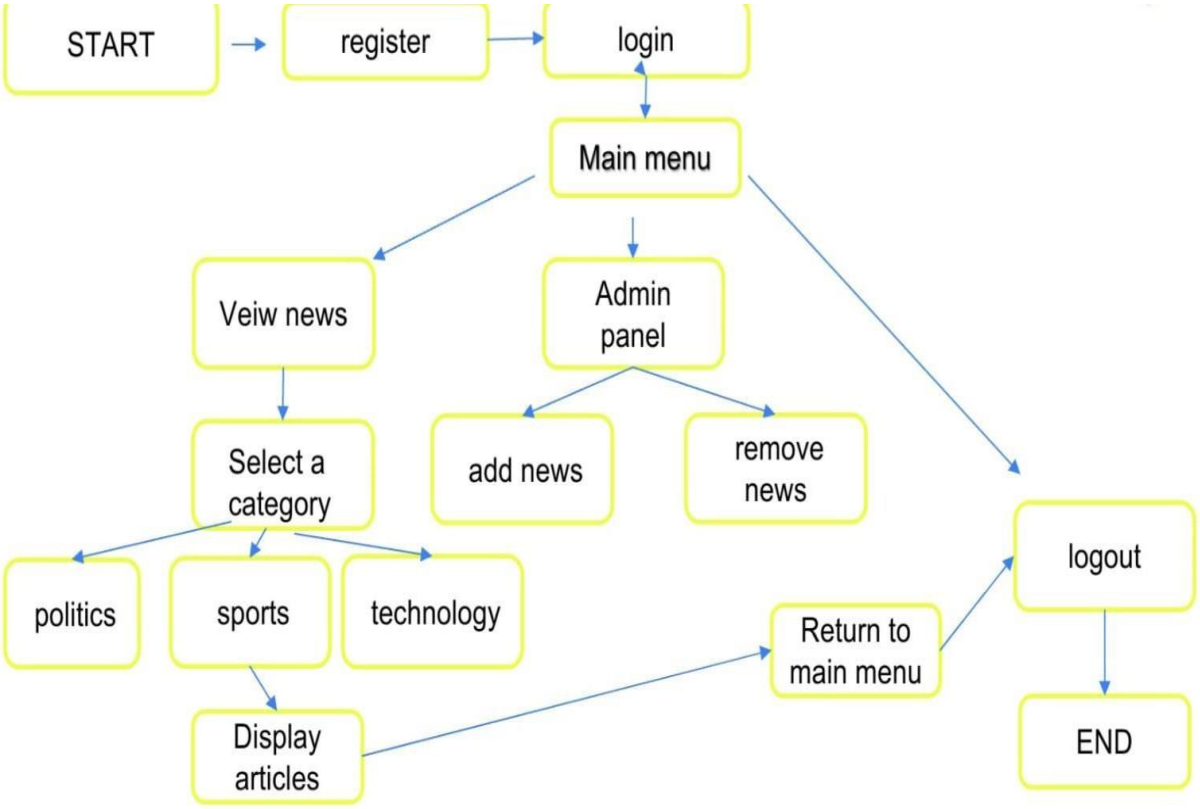
List: A GUI component used for displaying articles.

CHAPTER 2
PROJECT METHODOLOGY

2.1 Proposed Work

The proposed work for the News Portal application focuses on enhancing its functionality, security, and user experience. One key improvement is integrating a database, such as MySQL or SQLite, to persist user data, roles, and articles, ensuring data retention across sessions. To enhance security, password hashing should be implemented for safe storage of user credentials, along with a password strength validation feature. Admin functionalities can be expanded to include the ability to edit and update articles, as well as search articles by various parameters. The user interface can be modernized, potentially using JavaFX for a more attractive and responsive design. Additionally, a user profile management system can be introduced, allowing users to update their information and bookmark articles. Role-based permissions should be further refined to support advanced access control for Admins and Users. Interaction features such as article comments, likes, and a notification system could be added to increase engagement. Finally, the application would benefit from rigorous testing, error handling improvements, and the implementation of a more scalable architecture, which together would make the portal more secure, user-friendly, and robust.

2.2 Block Diagram



CHAPTER 3
MODULE DESCRIPTION

3.1 User Authentication and Registration Module:

This module handles the login and registration functionality. It allows users to register with a username, password, and role (either Admin or User). During login, the system verifies the credentials against stored user data. If the credentials are valid, users are granted access to the appropriate dashboard (Admin or User). This module ensures secure user access and registration**.**

3.2 User Dashboard Module:

The User Dashboard provides users with the ability to view and interact with news articles. It allows users to browse articles from different categories (such as Politics and Sports) and read article details. Users can select articles, which will open a dialog displaying the content. The dashboard is also responsible for allowing users to log out of the system and return to the login screen.

3.3 Admin Management Module:

This module provides Admins with the tools to manage the portal's content. Admins can add new articles, edit existing ones, and delete articles across categories. The Admin Management module includes functionality to select categories, view articles within those categories, and manage article details (heading and context). Admins also have the ability to navigate between various panels, including the user dashboard.

3.4 Article Management Module:

This module focuses on handling articles within the portal. It allows Admins to add new articles to predefined categories (e.g., Politics, Sports), store the articles, and remove articles when necessary. The articles are stored in memory using an ArrayList for each category. This module ensures articles are properly categorized and easily accessible for both Admins and Users.

3.5 User Interface and Event Handling Module:

This module manages the graphical user interface (GUI) and user interaction with the portal. It uses AWT components like buttons, labels, text fields, and lists to create an interactive experience. Event listeners, such as ActionListener and ItemListener, are implemented to handle user actions like logging in, registering, selecting articles, and navigating between different panels. This module ensures a smooth user experience with an intuitive interface and responsive event handling.

CHAPTER 4
CONCLUSION & FUTURE SCOPE

4.1 CONCLUSION

In conclusion, the News Portal application provides a comprehensive platform for both users and administrators to interact with news content efficiently. By implementing distinct modules for user authentication, content management, and user interface handling, the application ensures a seamless experience. The system's role-based access control allows for secure access and flexible management of content, making it suitable for various user types. While the current implementation is functional, there are several opportunities for improvement, such as database integration, enhanced security, and additional user interaction features. Future development can focus on making the application more robust, scalable, and user-friendly. With these enhancements, the News Portal can become a powerful and professional tool for managing and consuming news content. Overall, the application lays a strong foundation for further growth and feature expansion.

4.2 FUTURE SCOPE

The future scope of the News Portal application offers numerous opportunities for growth and enhancement. One of the key areas for future development is the integration of a database, which would enable persistent storage of user data, articles, and preferences, ensuring the system remains consistent across sessions. Additionally, implementing more advanced security features, such as password encryption and multi-factor authentication, would significantly improve the application's security. Expanding the user interaction capabilities, such as adding comments, likes, and a notification system, would increase user engagement. The addition of an advanced search and filtering system would enhance content discovery for both users and administrators. Moreover, integrating a more modern and responsive user interface, possibly with JavaFX, would improve the overall user experience across various devices and screen sizes. The application could also evolve

to support multi-language content and allow more customization for users to tailor their experience. Lastly, implementing machine learning algorithms to suggest articles based on user preferences could make the portal more dynamic and personalized. These enhancements would not only make the platform more feature- rich but also more scalable and capable of meeting the needs of a broader audience.

APPENDIX A

```
import java.awt.*;
import java.awt.event.*;
import java.util.ArrayList;
import java.util.HashMap;

public class NewsPortal extends Frame implements ActionListener {
   private CardLayout cardLayout;
   private Panel cards;
   private TextField usernameField, passwordField, regUsernameField, regPasswordField;
   private Choice roleChoice, categoryChoice, adminCategoryChoice;
   private List newsList, adminNewsList;
   private TextField articleHeadingField;
   private TextArea articleContextArea;

   private HashMap<String, String> users; // Username -> Password
   private HashMap<String, String> roles; // Username -> Role
   private HashMap<String, ArrayList<Article>> articles; // Category -> List of Articles

   private String currentUsername = null;

   // Inner class for Articles
   static class Article {
      String heading;
      String context;

      public Article(String heading, String context) {
         this.heading = heading;
         this.context = context;
      }

      @Override
      public String toString() {
         return heading;
      }
```

```
}

public NewsPortal() {
    super("News Portal");
    users = new HashMap<>();
    roles = new HashMap<>();
    articles = new HashMap<>();
    articles.put("Politics", new ArrayList<>());
    articles.put("Sports", new ArrayList<>());

    cardLayout = new CardLayout();
    cards = new Panel(cardLayout);

    // Login Screen
    cards.add(createLoginPanel(), "Login");

    // Registration Screen
    cards.add(createRegistrationPanel(), "Register");

    // User Dashboard
    cards.add(createUserDashboard(), "Dashboard");

    // Admin Panel
    cards.add(createAdminPanel(), "Admin");

    add(cards);
    setSize(800, 500);
    setVisible(true);
    setBackground(new Color(200, 220, 255));
    addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            dispose();
        }
    });
}

private Panel createLoginPanel() {
    Panel loginPanel = new Panel(new GridBagLayout());
```

```
GridBagConstraints gbc = new GridBagConstraints();
gbc.insets = new Insets(10, 10, 10, 10);

loginPanel.setBackground(new Color(230, 230, 250));

gbc.gridx = 0;
gbc.gridy = 0;
gbc.gridwidth = 2;
Label title = new Label("News Portal", Label.CENTER);
title.setFont(new Font("Arial", Font.BOLD, 24));
loginPanel.add(title, gbc);

gbc.gridy = 1;
gbc.gridwidth = 1;
loginPanel.add(new Label("Username:"), gbc);

gbc.gridx = 1;
usernameField = new TextField(20);
loginPanel.add(usernameField, gbc);

gbc.gridx = 0;
gbc.gridy = 2;
loginPanel.add(new Label("Password:"), gbc);

gbc.gridx = 1;
passwordField = new TextField(20);
passwordField.setEchoChar('*');
loginPanel.add(passwordField, gbc);

Panel buttonPanel = new Panel(new FlowLayout());
Button loginButton = new Button("Login");
loginButton.setBackground(new Color(144, 238, 144));
loginButton.addActionListener(this);
buttonPanel.add(loginButton);

Button registerButton = new Button("Register");
registerButton.setBackground(new Color(173, 216, 230));
registerButton.addActionListener(this);
```

```
    buttonPanel.add(registerButton);

    gbc.gridx = 0;
    gbc.gridy = 3;
    gbc.gridwidth = 2;
    loginPanel.add(buttonPanel, gbc);

    return loginPanel;
}

private Panel createRegistrationPanel() {
    Panel registerPanel = new Panel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(10, 10, 10, 10);

    registerPanel.setBackground(new Color(245, 245, 220));

    gbc.gridx = 0;
    gbc.gridy = 0;
    registerPanel.add(new Label("New Username:"), gbc);

    gbc.gridx = 1;
    regUsernameField = new TextField(20);
    registerPanel.add(regUsernameField, gbc);

    gbc.gridx = 0;
    gbc.gridy = 1;
    registerPanel.add(new Label("New Password:"), gbc);

    gbc.gridx = 1;
    regPasswordField = new TextField(20);
    regPasswordField.setEchoChar('*');
    registerPanel.add(regPasswordField, gbc);

    gbc.gridx = 0;
    gbc.gridy = 2;
    registerPanel.add(new Label("Role:"), gbc);
```

```
    gbc.gridx = 1;
    roleChoice = new Choice();
    roleChoice.add("User");
    roleChoice.add("Admin");
    registerPanel.add(roleChoice, gbc);

    Panel buttonPanel = new Panel(new FlowLayout());
    Button regSubmitButton = new Button("Submit");
    regSubmitButton.setBackground(new Color(144, 238, 144));
    regSubmitButton.addActionListener(this);
    buttonPanel.add(regSubmitButton);

    Button regBackButton = new Button("Back");
    regBackButton.setBackground(new Color(173, 216, 230));
    regBackButton.addActionListener(this);
    buttonPanel.add(regBackButton);

    gbc.gridx = 0;
    gbc.gridy = 3;
    gbc.gridwidth = 2;
    registerPanel.add(buttonPanel, gbc);

    return registerPanel;
}

private Panel createUserDashboard() {
    Panel userDashboard = new Panel(new BorderLayout());
    userDashboard.setBackground(new Color(240, 255, 255));

    categoryChoice = new Choice();
    for (String category : articles.keySet()) {
        categoryChoice.add(category);
    }
    categoryChoice.addItemListener(e -> updateNewsList());
    userDashboard.add(categoryChoice, BorderLayout.NORTH);

    newsList = new List();
    newsList.addItemListener(e -> handleNewsListClick());
```

```
    userDashboard.add(newsList, BorderLayout.CENTER);

    Button logoutButton = new Button("Logout");
    logoutButton.setBackground(new Color(250, 128, 114));
    logoutButton.addActionListener(this);
    userDashboard.add(logoutButton, BorderLayout.SOUTH);

    return userDashboard;
}

private Panel createAdminPanel() {
    Panel adminPanel = new Panel(new GridBagLayout());
    GridBagConstraints gbc = new GridBagConstraints();
    gbc.insets = new Insets(10, 10, 10, 10);

    adminPanel.setBackground(new Color(255, 239, 213));

    gbc.gridx = 0;
    gbc.gridy = 0;
    adminPanel.add(new Label("Category:"), gbc);

    gbc.gridx = 1;
    adminCategoryChoice = new Choice();
    for (String category : articles.keySet()) {
        adminCategoryChoice.add(category);
    }
    adminCategoryChoice.addItemListener(e -> updateAdminNewsList());
    adminPanel.add(adminCategoryChoice, gbc);

    gbc.gridx = 0;
    gbc.gridy = 1;
    adminPanel.add(new Label("Articles:"), gbc);

    gbc.gridx = 1;
    adminNewsList = new List();
    adminPanel.add(adminNewsList, gbc);

    gbc.gridx = 0;
```

```
        gbc.gridy = 2;
        adminPanel.add(new Label("Heading:"), gbc);

        gbc.gridx = 1;
        articleHeadingField = new TextField(20);
        adminPanel.add(articleHeadingField, gbc);

        gbc.gridx = 0;
        gbc.gridy = 3;
        adminPanel.add(new Label("Context:"), gbc);

        gbc.gridx = 1;
        articleContextArea = new TextArea(5, 20);
        adminPanel.add(articleContextArea, gbc);

        Panel buttonPanel = new Panel(new FlowLayout());
        Button addArticleButton = new Button("Add Article");
        addArticleButton.setBackground(new Color(144, 238, 144));
        addArticleButton.addActionListener(e -> handleAddArticle());
        buttonPanel.add(addArticleButton);

        Button deleteArticleButton = new Button("Delete Article");
        deleteArticleButton.setBackground(new Color(255, 99, 71));
        deleteArticleButton.addActionListener(e -> handleDeleteArticle());
        buttonPanel.add(deleteArticleButton);

        Button adminBackButton = new Button("Back");
        adminBackButton.setBackground(new Color(173, 216, 230));
        adminBackButton.addActionListener(e -> cardLayout.show(cards,
"Dashboard"));
        buttonPanel.add(adminBackButton);

        gbc.gridx = 0;
        gbc.gridy = 4;
        gbc.gridwidth = 2;
        adminPanel.add(buttonPanel, gbc);

        return adminPanel;
```

```
    }

    private void updateAdminNewsList() {
        String selectedCategory = adminCategoryChoice.getSelectedItem();
        adminNewsList.removeAll();
        for (Article article : articles.getOrDefault(selectedCategory, new ArrayList<>()))
{
            adminNewsList.add(article.heading);
        }
    }

    private void updateNewsList() {
        String selectedCategory = categoryChoice.getSelectedItem();
        newsList.removeAll();
        for (Article article : articles.getOrDefault(selectedCategory, new ArrayList<>()))
{
            newsList.add(article.heading);
        }
    }

    private void handleNewsListClick() {
        String selectedHeading = newsList.getSelectedItem();
        String selectedCategory = categoryChoice.getSelectedItem();

        for (Article article : articles.getOrDefault(selectedCategory, new ArrayList<>()))
{
            if (article.heading.equals(selectedHeading)) {
                showArticleDetailsDialog(article);
                return;
            }
        }
    }

    private void handleDeleteArticle() {
        String selectedCategory = adminCategoryChoice.getSelectedItem();
        String selectedHeading = adminNewsList.getSelectedItem();

        ArrayList<Article> categoryArticles = articles.getOrDefault(selectedCategory,
```

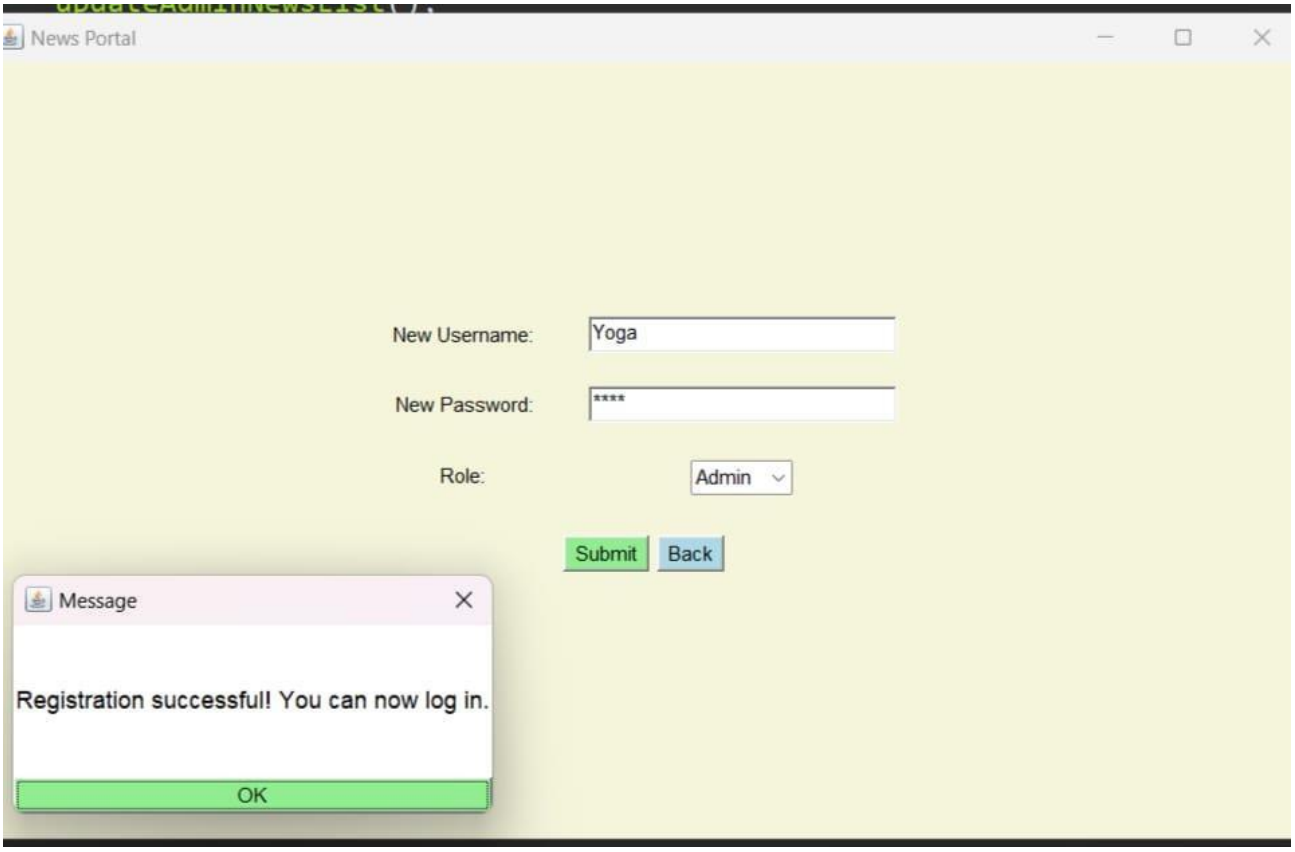
```
new ArrayList<>());
        categoryArticles.removeIf(article -> article.heading.equals(selectedHeading));
        updateAdminNewsList();
        showMessageDialog("Article deleted successfully.");
    }

    private void showArticleDetailsDialog(Article article) {
        Dialog dialog = new Dialog(this, "Article Details", true);
        dialog.setLayout(new BorderLayout());

        Label headingLabel = new Label(article.heading, Label.CENTER);
        headingLabel.setFont(new Font("Arial", Font.BOLD, 20));
        dialog.add(headingLabel, BorderLayout.NORTH);

        TextArea contextArea = new TextArea(article.context, 10, 50,
TextArea.SCROLLBARS_VERTICAL_ONLY);
        contextArea.setEditable(false);
        contextArea.setFont(new Font("Arial", Font.PLAIN, 14));
        dialog.add(contextArea, BorderLayout.CENTER);

        Button closeButton = new Button("Close");
        closeButton.addActionListener(e -> dialog.dispose());
        dialog.add(closeButton, BorderLayout.SOUTH);

        dialog.setSize(400, 300);
        dialog.setVisible(true);
    }

    private void handleAddArticle() {
        String category = adminCategoryChoice.getSelectedItem();
        String heading = articleHeadingField.getText().trim();
        String context = articleContextArea.getText().trim();

        if (heading.isEmpty() || context.isEmpty()) {
            showMessageDialog("Heading and context cannot be empty!");
            return;
        }
```

```
    Article newArticle = new Article(heading, context);
    articles.get(category).add(newArticle);
    updateAdminNewsList();
    showMessageDialog("Article added successfully to " + category + "!");
    articleHeadingField.setText("");
    articleContextArea.setText("");
}

public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();
    switch (command) {
        case "Login":
            handleLogin();
            break;
        case "Register":
            cardLayout.show(cards, "Register");
            break;
        case "Submit":
            handleRegistration();
            break;
        case "Back":
            cardLayout.show(cards, "Login");
            break;
        case "Logout":
            currentUsername = null;
            cardLayout.show(cards, "Login");
            break;
    }
}

private void handleLogin() {
    String username = usernameField.getText();
    String password = passwordField.getText();

    if (users.containsKey(username) && users.get(username).equals(password)) {
        currentUsername = username;
        String role = roles.get(username);
        if ("Admin".equals(role)) {
```

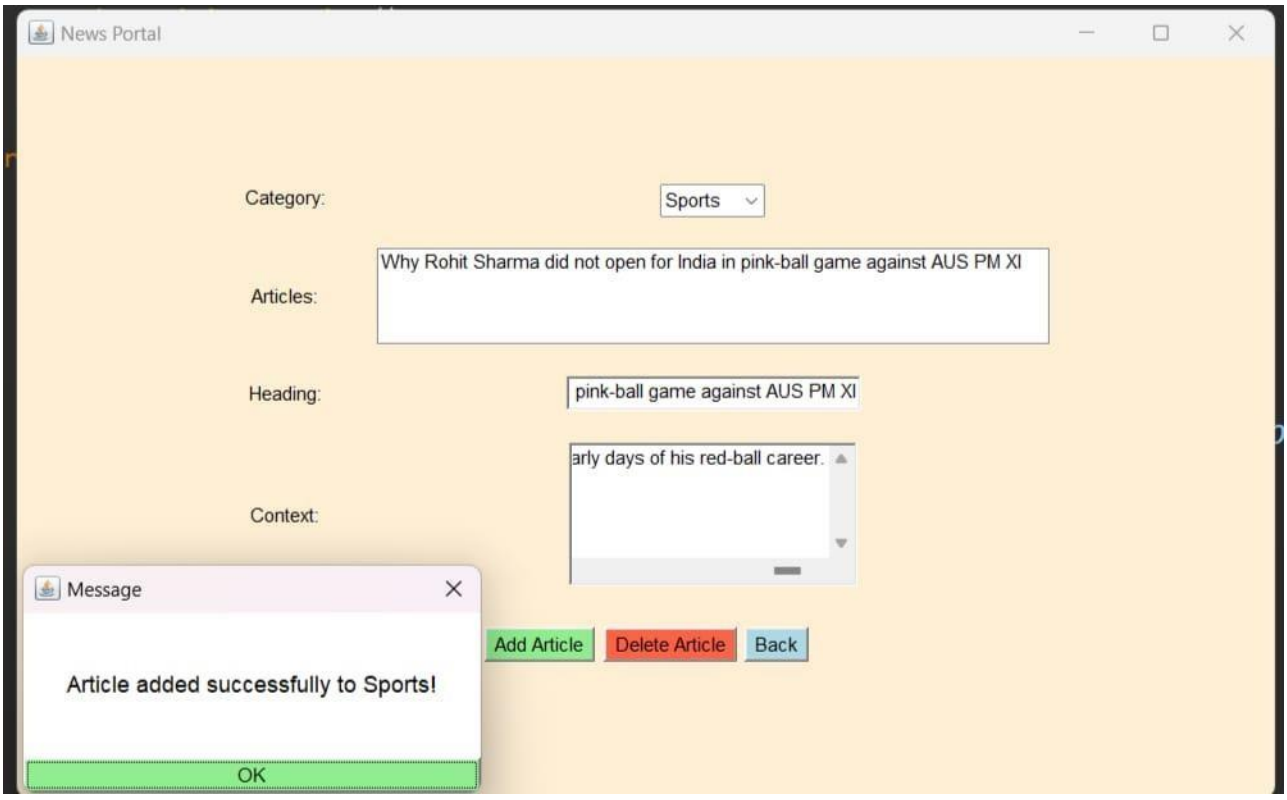
```
                cardLayout.show(cards, "Admin");
            } else {
                cardLayout.show(cards, "Dashboard");
            }
        } else {
            showMessageDialog("Invalid username or password!");
        }
    }

    private void handleRegistration() {
        String username = regUsernameField.getText();
        String password = regPasswordField.getText();
        String role = roleChoice.getSelectedItem();

        if (users.containsKey(username)) {
            showMessageDialog("Username already exists. Choose a different
username.");
        } else if (username.isEmpty() || password.isEmpty()) {
            showMessageDialog("Username and password cannot be empty.");
        } else {
            users.put(username, password);
            roles.put(username, role);
            showMessageDialog("Registration successful! You can now log in.");
            cardLayout.show(cards, "Login");
        }
    }

    private void showMessageDialog(String message) {
        Dialog dialog = new Dialog(this, "Message", true);
        dialog.setLayout(new BorderLayout());
        dialog.setBackground(Color.WHITE);

        Label messageLabel = new Label(message, Label.CENTER);
        messageLabel.setFont(new Font("Arial", Font.PLAIN, 14));
        dialog.add(messageLabel, BorderLayout.CENTER);

        Button okButton = new Button("OK");
        okButton.setBackground(new Color(144, 238, 144));
```

```
        okButton.addActionListener(e -> dialog.dispose());
        dialog.add(okButton, BorderLayout.SOUTH);

        dialog.setSize(300, 150);
        dialog.setVisible(true);
    }

    public static void main(String[] args) {
        new NewsPortal();
    }
}
```

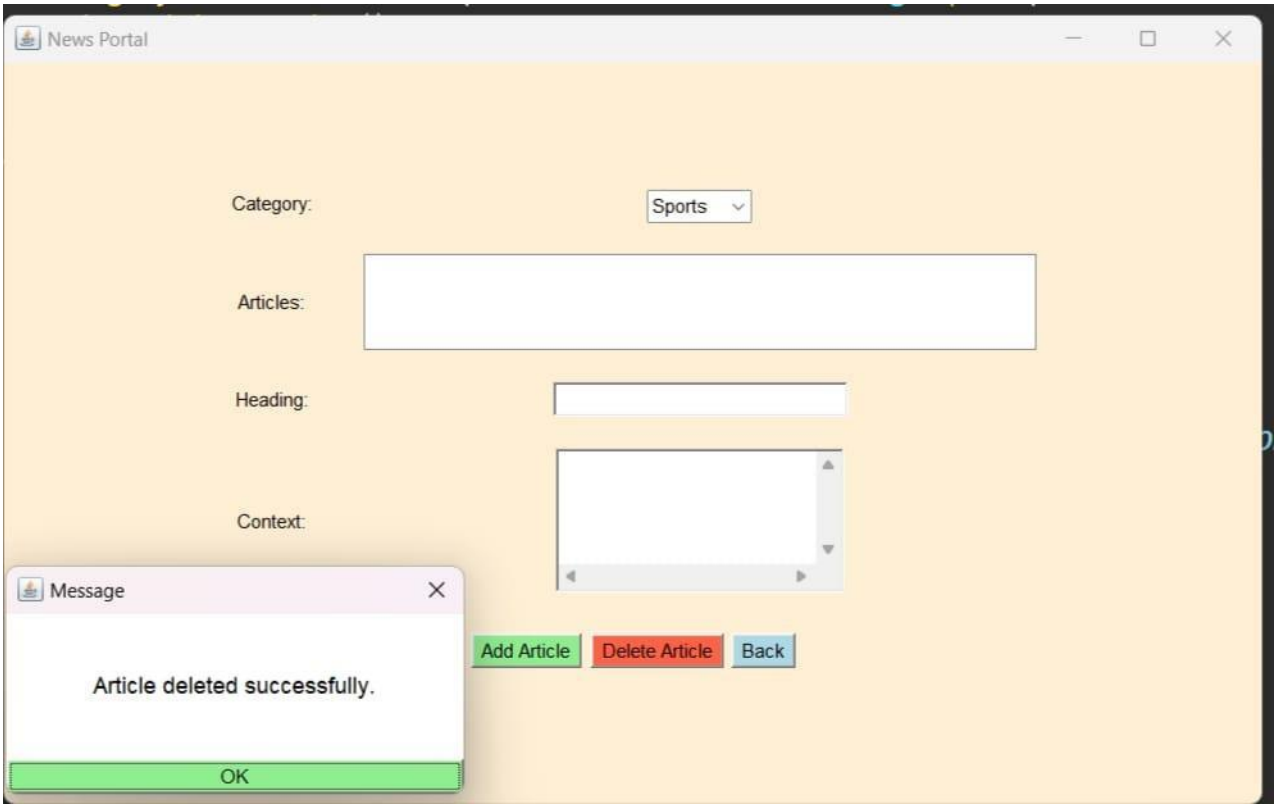
APPENDIX B


News Portal
New Username:
Yoga
New Password:
****
Role:
Admin
Submit
Back
Message
Registration successful! You can now log in.
OK


News Portal
News Portal
Username:
Yoga
Password:
****
Login
Register


News Portal
Category:
Sports
Articles:
Why Rohit Sharma did not open for India in pink-ball game against AUS PM XI
Heading:
pink-ball game against AUS PM XI
Context:
arly days of his red-ball career.
Message
Article added successfully to Sports!
OK
Add Article
Delete Article
Back


News Portal
Sports
Why Rohit Sharma did not open for India in pink-ball game against AUS PM XI
Article Details
Why Rohit Sharma did not open for India in pink-ball game against AUS PM XI
Rohit listed himself at 5 on the team sheet after he won the toss and opted to field. It was expected that after the break after the New Zealand series, he would look to use this game to adapt to conditions ahead of the Day-Night Test in Adelaide starting from December 6. With Shubman Gill also likely to return, India could be experimenting with a different batting position for the captain, one he is all too familiar with from the early days of his red-ball career.
Close
Logout


News Portal
Category:
Sports
Articles:
Heading:
Context:
Add Article
Delete Article
Back
Message
Article deleted successfully.
OK

REFERENCES

1. Java AWT (Abstract Window Toolkit): Oracle's official documentation on AWT components: AWT Documentation
2. Java Event Handling:Understanding event handling in Java, including the use of ActionListener and ItemListener: Event Handling in Java
3. Role-Based Access Control (RBAC) in Java: General overview of RBAC and its implementation patterns for web or desktop applications: RBAC Overview