```java
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import java.util.Scanner;

public class NewsPortalWithCategories extends Frame implements ActionListener {
   private TextField titleField, usernameField, passwordField;
   private TextArea contentArea, displayArea;
   private Choice categoryChoice, filterChoice;
   private Button saveButton, viewButton, filterButton, clearButton, loginButton;
   private Label statusLabel;
   private Panel loginPanel, portalPanel;

   private static final String USERNAME = "admin";
   private static final String PASSWORD = "admin123";
   private static final String FILE_NAME = "articles_with_categories.txt";

   public NewsPortalWithCategories() {
      // Set up the frame
      setTitle("News Portal System with Categories");
      setSize(600, 700);
      setLayout(new CardLayout());
      setResizable(false);

      // Login Panel
      loginPanel = new Panel(new GridLayout(4, 2));
      loginPanel.add(new Label("Username:"));
      usernameField = new TextField();
      loginPanel.add(usernameField);

      loginPanel.add(new Label("Password:"));
      passwordField = new TextField();
      passwordField.setEchoChar('*');
      loginPanel.add(passwordField);

      loginButton = new Button("Login");
      loginButton.addActionListener(this);
      loginPanel.add(loginButton);

      statusLabel = new Label("");
      statusLabel.setForeground(Color.RED);
      loginPanel.add(statusLabel);

      add(loginPanel, "Login");
```

```
// Portal Panel
portalPanel = new Panel(new FlowLayout());

portalPanel.add(new Label("Article Title:"));
titleField = new TextField(40);
portalPanel.add(titleField);

portalPanel.add(new Label("Category:"));
categoryChoice = new Choice();
categoryChoice.add("Sports");
categoryChoice.add("Technology");
categoryChoice.add("Normal News");
portalPanel.add(categoryChoice);

portalPanel.add(new Label("Article Content:"));
contentArea = new TextArea(10, 50);
portalPanel.add(contentArea);

saveButton = new Button("Save Article");
saveButton.addActionListener(this);
portalPanel.add(saveButton);

viewButton = new Button("View All Articles");
viewButton.addActionListener(this);
portalPanel.add(viewButton);

portalPanel.add(new Label("Filter by Category:"));
filterChoice = new Choice();
filterChoice.add("All");
filterChoice.add("Sports");
filterChoice.add("Technology");
filterChoice.add("Normal News");
portalPanel.add(filterChoice);

filterButton = new Button("Filter Articles");
filterButton.addActionListener(this);
portalPanel.add(filterButton);

clearButton = new Button("Clear");
clearButton.addActionListener(this);
portalPanel.add(clearButton);

displayArea = new TextArea(20, 50);
```

```
        displayArea.setEditable(false);
        portalPanel.add(displayArea);

        // Add the portal panel but hide it initially
        portalPanel.setVisible(false);
        add(portalPanel, "Portal");

        // Window close handler
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == loginButton) {
            authenticateUser();
        } else if (e.getSource() == saveButton) {
            saveArticle();
        } else if (e.getSource() == viewButton) {
            viewAllArticles();
        } else if (e.getSource() == filterButton) {
            filterArticles();
        } else if (e.getSource() == clearButton) {
            clearInputs();
        }
    }

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

        if (USERNAME.equals(username) && PASSWORD.equals(password)) {
            statusLabel.setText("Login successful!");
            loginPanel.setVisible(false);
            portalPanel.setVisible(true);
        } else {
            statusLabel.setText("Invalid username or password.");
        }
    }

    private void saveArticle() {
        String title = titleField.getText().trim();
        String category = categoryChoice.getSelectedItem();
        String content = contentArea.getText().trim();

        if (title.isEmpty() || content.isEmpty()) {
            displayArea.setText("Title and content cannot be empty!");
            return;
        }

        try (FileWriter writer = new FileWriter(FILE_NAME, true)) {
            writer.write("Category: " + category + "\n");
            writer.write("Title: " + title + "\n");
            writer.write("Content: " + content + "\n");
            writer.write("=".repeat(50) + "\n");
            displayArea.setText("Article saved successfully!");
        } catch (IOException ex) {
            displayArea.setText("Error saving the article: " + ex.getMessage());
        }
    }

    private void viewAllArticles() {
        File file = new File(FILE_NAME);

        if (!file.exists()) {
            displayArea.setText("No articles have been saved yet.");
            return;
        }

        try (Scanner scanner = new Scanner(file)) {
            StringBuilder content = new StringBuilder();
            while (scanner.hasNextLine()) {
                content.append(scanner.nextLine()).append("\n");
            }
            displayArea.setText(content.toString());
        } catch (IOException ex) {
            displayArea.setText("Error reading articles: " + ex.getMessage());
        }
    }

    private void filterArticles() {
        String selectedCategory = filterChoice.getSelectedItem();
        File file = new File(FILE_NAME);

        if (!file.exists()) {
            displayArea.setText("No articles have been saved yet.");
            return;
        }

        try (Scanner scanner = new Scanner(file)) {
            StringBuilder content = new StringBuilder();
            while (scanner.hasNextLine()) {
                String line = scanner.nextLine();
                if (line.startsWith("Category: ")) {
                    String category = line.substring(10).trim();
                    if (selectedCategory.equals("All") || category.equals(selectedCategory)) {
                        content.append(line).append("\n");
                        while (scanner.hasNextLine()) {
                            String articleLine = scanner.nextLine();
                            content.append(articleLine).append("\n");
                            if (articleLine.equals("=".repeat(50))) {
                                break;
                            }
                        }
                    }
                }
            }
            displayArea.setText(content.toString());
        } catch (IOException ex) {
            displayArea.setText("Error filtering articles: " + ex.getMessage());
        }
    }

    private void clearInputs() {
        titleField.setText("");
        contentArea.setText("");
        displayArea.setText("");
    }

    public static void main(String[] args) {
        NewsPortalWithCategories portal = new NewsPortalWithCategories();
        portal.setVisible(true);
    }
}
```