package chefdonburi;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Image;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyAdapter;

import java.awt.event.KeyEvent;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import javax.imageio.ImageIO;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

import java.io.File;

import java.io.IOException;

import org.springframework.security.crypto.bcrypt.BCrypt;

public class Login implements ActionListener {

JFrame frmLogin;

JPanel pnlLogin;

JLabel lblUser, lblPassword, lblTitle;

JTextField txtUser;

JPasswordField pswPassword;

JButton btnLogin, btnClose;

// For database connection

Connection connection;

PreparedStatement ps;

ResultSet rs;

// Background image

private Image backgroundImage;

public Login() {

// Load the background image

try {

backgroundImage = ImageIO.read(new File("src\\Images\\login\_background.png"));

} catch (IOException e) {

e.printStackTrace();

}

}

public void init() {

ImageIcon frameicon = new ImageIcon("src\\Images\\jframeicon.jpg");

Image frame = frameicon.getImage().getScaledInstance(100, 100, Image.SCALE\_SMOOTH);

frmLogin = new JFrame("Chef Donburi Login");

frmLogin.setIconImage(frame);

frmLogin.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frmLogin.setSize(1266, 668);

frmLogin.setLocationRelativeTo(null);

pnlLogin = new JPanel() {

@Override

protected void paintComponent(Graphics g) {

super.paintComponent(g);

if (backgroundImage != null) {

g.drawImage(backgroundImage, 0, 0, getWidth(), getHeight(), this);

}

}

};

pnlLogin.setLayout(null);

// Title label

lblTitle = new JLabel("Welcome");

lblTitle.setForeground(new Color(242, 245, 224));

lblTitle.setFont(new Font("Georgia", Font.BOLD, 40));

lblTitle.setSize(250, 50);

lblTitle.setLocation(925, 200);

pnlLogin.add(lblTitle);

// Username label and text field

lblUser = new JLabel("Username:");

lblUser.setForeground(new Color(242, 245, 224));

lblUser.setFont(new Font("Arial", Font.BOLD, 20));

lblUser.setSize(200, 25);

lblUser.setLocation(780, 270);

pnlLogin.add(lblUser);

txtUser = new JTextField();

txtUser.setSize(250, 25);

txtUser.setLocation(900, 270);

pnlLogin.add(txtUser);

// Password label and password field

lblPassword = new JLabel("Password:");

lblPassword.setForeground(new Color(242, 245, 224));

lblPassword.setFont(new Font("Arial", Font.BOLD, 20));

lblPassword.setSize(200, 25);

lblPassword.setLocation(780, 320);

pnlLogin.add(lblPassword);

pswPassword = new JPasswordField();

pswPassword.setSize(250, 25);

pswPassword.setLocation(900, 320);

pnlLogin.add(pswPassword);

// Login button

ImageIcon loginicon = new ImageIcon("src\\Images\\login.png");

Image login = loginicon.getImage().getScaledInstance(25, 25, Image.SCALE\_SMOOTH);

btnLogin = new JButton("Login", new ImageIcon(login));

btnLogin.setBackground(new Color(242, 245, 224));

btnLogin.setSize(250, 30);

btnLogin.setLocation(900, 370);

btnLogin.addActionListener(this);

pnlLogin.add(btnLogin);

// Close button

ImageIcon closeicon = new ImageIcon("src\\Images\\close.png");

Image close = closeicon.getImage().getScaledInstance(20, 20, Image.SCALE\_SMOOTH);

btnClose = new JButton("Close", new ImageIcon(close));

btnClose.setBackground(new Color(242, 245, 224));

btnClose.setSize(250, 30);

btnClose.setLocation(900, 420);

btnClose.addActionListener(this);

pnlLogin.add(btnClose);

// Add KeyListener to handle "Enter" key press

txtUser.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_ENTER) {

loginUser(); // Trigger login action when Enter is pressed

}

}

});

pswPassword.addKeyListener(new KeyAdapter() {

@Override

public void keyPressed(KeyEvent e) {

if (e.getKeyCode() == KeyEvent.VK\_ENTER) {

loginUser(); // Trigger login action when Enter is pressed

}

}

});

frmLogin.add(pnlLogin);

frmLogin.setVisible(true);

}

@Override

public void actionPerformed(ActionEvent e) {

if (e.getSource() == btnLogin) {

loginUser(); // Handle login

} else if (e.getSource() == btnClose) {

int response = JOptionPane.showConfirmDialog(frmLogin, "Are you sure you want to close?", "Confirm Close", JOptionPane.YES\_NO\_OPTION);

if (response == JOptionPane.YES\_OPTION) {

frmLogin.dispose(); // Close the application

}

}

}

private void loginUser() {

String username = txtUser.getText();

String password = new String(pswPassword.getPassword());

if (username.isEmpty() || password.isEmpty()) {

JOptionPane.showMessageDialog(frmLogin, "Username or Password cannot be empty.");

return;

}

try {

connection = new Database().getConnection();

String query = "SELECT \* FROM users WHERE userName = ?";

ps = connection.prepareStatement(query);

ps.setString(1, username);

rs = ps.executeQuery();

if (rs.next()) {

String storedPasswordHash = rs.getString("userPassword");

String role = rs.getString("role");

int userID = rs.getInt("userID");

// Verify password using BCrypt

if (BCrypt.checkpw(password, storedPasswordHash)) {

JOptionPane.showMessageDialog(frmLogin, "Login successful!");

logUserLogin(userID);

frmLogin.dispose();

Home home = new Home(role, userID); // Open home screen

} else {

JOptionPane.showMessageDialog(frmLogin, "Invalid username or password.");

}

} else {

JOptionPane.showMessageDialog(frmLogin, "Invalid username or password.");

}

} catch (SQLException ex) {

JOptionPane.showMessageDialog(frmLogin, "Error connecting to database: " + ex.getMessage());

} finally {

closeConnections();

}

}

private void logUserLogin(int userID) {

try {

String query = "INSERT INTO logs (userID, loginTime) VALUES (?, NOW())";

ps = connection.prepareStatement(query);

ps.setInt(1, userID);

ps.executeUpdate();

} catch (SQLException ex) {

JOptionPane.showMessageDialog(frmLogin, "Error logging login event: " + ex.getMessage());

}

}

private void closeConnections() {

try {

if (rs != null) rs.close();

if (ps != null) ps.close();

if (connection != null) connection.close();

} catch (SQLException e) {

JOptionPane.showMessageDialog(frmLogin, "Error closing resources: " + e.getMessage());

}

}

public static void main(String[] args) {

new Login().init();

}

}