JAVA PROJECT REPORT

(Project Term January-May 2023)

Lockr: Protect Your Digital Identity with a Password Manager

Submitted by

Name: Rohan Dilip Nair Registration Number:12110392

COURSE CODE: CSE 310

Under the Guidance of

Dr. Ranjith Kumar A

School of Computer Science and Engineering

LOVELY PROFESSIONAL UNIVERSITY

DECLARATION

I hereby declare that the project work entitled (“**Lockr: Protect Your Digital Identity with a Password Manager**”) is an authentic record of our own work carried out as requirements of Capstone Project for the award of B.Tech degree in Computer Science Engineering from Lovely Professional University, Phagwara, under the guidance of (Dr. Ranjith Kumar A), during January to May 2023. All the information furnished in this capstone project report is based on my own intensive work and is genuine.

Name of Student:

Rohan Dilip Nair

Registration Number:

12110392

Roll Number:

K21STB63

(Signature of Student)
Rohan Dilip Nair

Date:
23/04/2023

TABLE OF CONTENTS

1. INTRODUCTION

A Password Manager is an application to help users remember their passwords by storing them so that the user can easily retrieve them when needed. The application starts with a login page. The program stores the username and password in a text file. Once logged in the user can choose to view, add, or generate passwords. The change password feature allows the user to update their login credentials. Overall, this program provides a basic implementation of a password manager.

1.1 Background of the Password Managers

Passwords are a fundamental aspect of digital security. The use of passwords became popular with the rise of the internet, as users needed to create accounts and authenticate their identity for online services. However, with the increase in the number of online services and accounts, it became challenging to create, manage and remember unique, strong passwords for each one. Password managers were developed to address these challenges, and they have become an essential tool for digital security.

Password managers are software applications designed to store and manage user's passwords securely. They allow users to create complex and unique passwords for each service, reducing the risk of a data breach or cyber-attack. Password managers encrypt user's passwords, so even if the password database is stolen, the attacker cannot read them without the encryption key. In addition, many password managers have built-in password generators, which can generate strong passwords for users automatically.

Password managers have a long history, with some early examples dating back to the 1990s. However, the widespread adoption of password managers came in the early 2000s with the development of cloud-based password managers. This was a significant improvement over earlier password managers, which required users to carry a physical device with them to access their passwords.

Today, password managers are available in many forms, including cloud-based, desktop, mobile, and browser-based. They offer a range of features, such as password auditing, two-factor authentication, and secure sharing of passwords with others. Many password managers are available for free, while others require a subscription fee. Overall, password managers have become a vital tool for online security, providing an easy and secure way for users to manage their passwords across multiple services.

1.2 OBJECTIVE OF THE PROJECT

1. Create a password management system.

2. Allow users to login to the system with a unique username and password.

3. To allow users to change their username and password.

4. Provide an interface for users to view, generate and add passwords to the password management system.

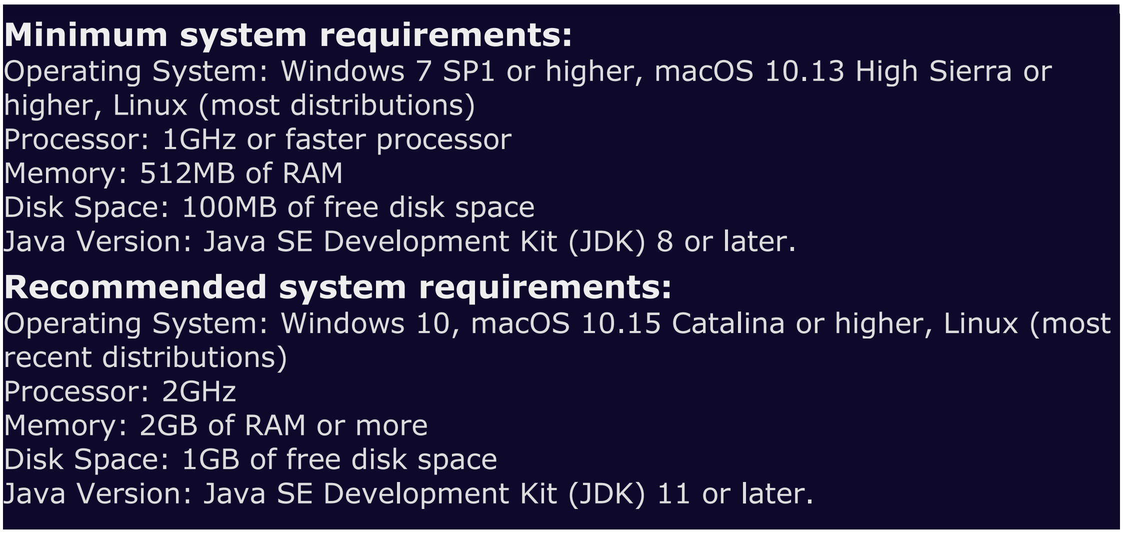
5. Use file handling to store and retrieve user login credentials and password information.

6. Implement a graphical user interface (GUI) for ease of use.

1.3 PROBLEM STATEMENT

- To ensure security, users require a reliable way to store multiple passwords. Password managers provide encrypted storage with a strong master password, guarding against unauthorized access.
- A password manager with the ability to generate random passwords is necessary to ensure unique and complex passwords for each account, with options to customize length, character types, and complexity.
- The user should have the ability to change stored passwords easily, with password manager prompting weak passwords to ensure security. Reusing or creating similar passwords should be avoided.

1.4 SYSTEM SPECIFICATIONS



Minimum system requirements:

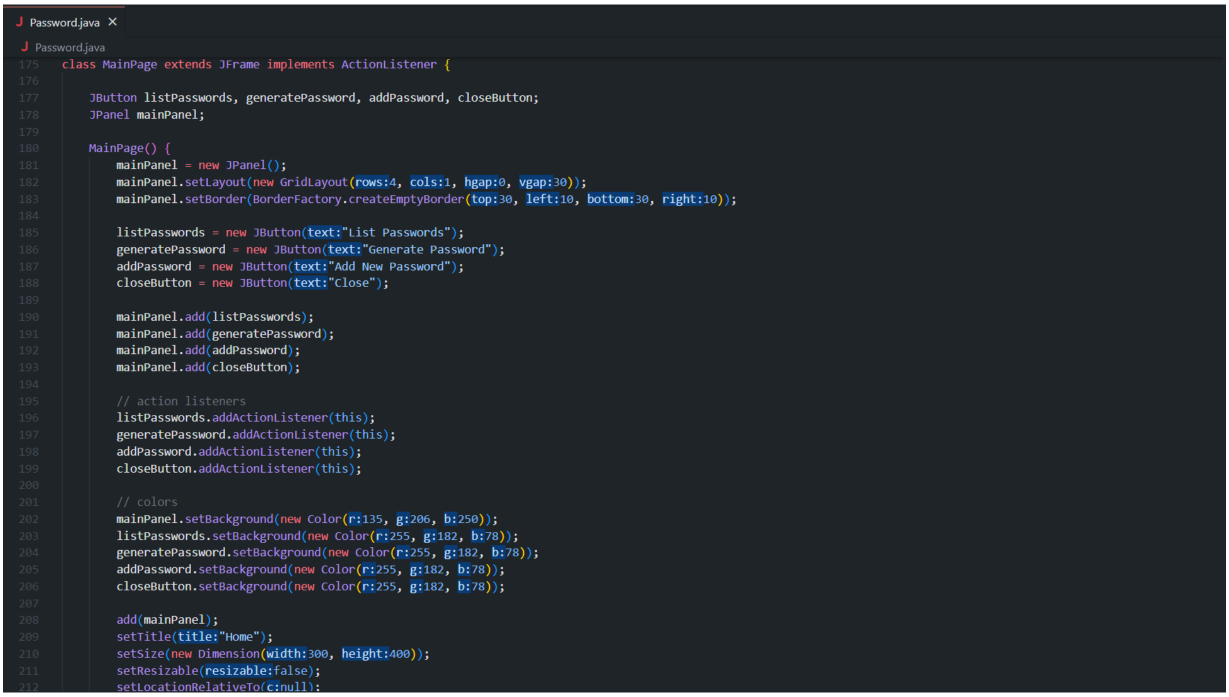
Operating System: Windows 7 SP1 or higher, macOS 10.13 High Sierra or higher, Linux (most distributions)
Processor: 1GHz or faster processor
Memory: 512MB of RAM
Disk Space: 100MB of free disk space
Java Version: Java SE Development Kit (JDK) 8 or later.

Recommended system requirements:

Operating System: Windows 10, macOS 10.15 Catalina or higher, Linux (most recent distributions)
Processor: 2GHz
Memory: 2GB of RAM or more
Disk Space: 1GB of free disk space
Java Version: Java SE Development Kit (JDK) 11 or later.

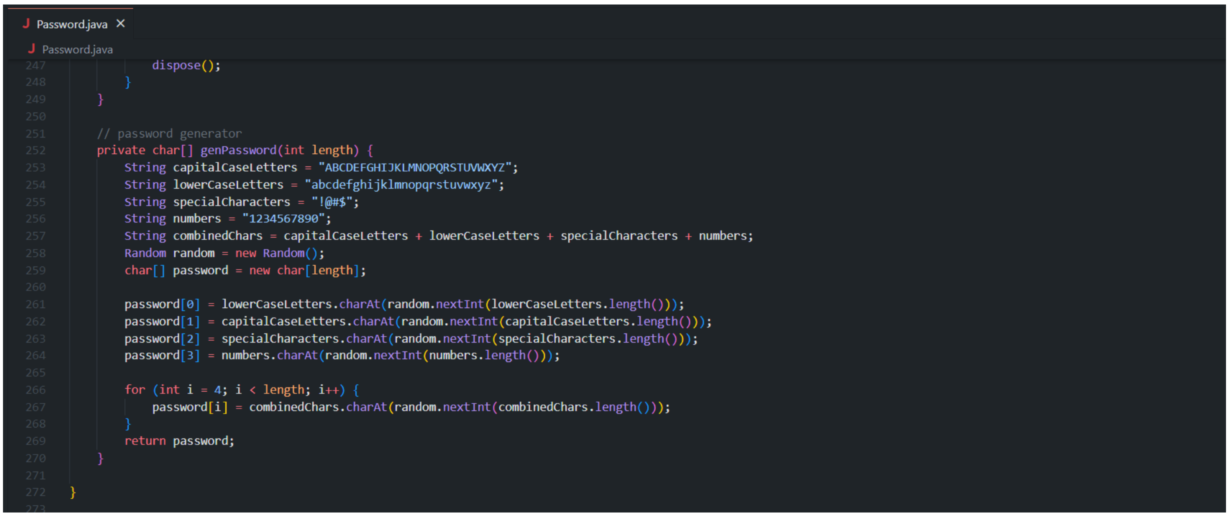
2. Module 1 (SCREENSHOTS OF THE CODE)

2.1 MAIN MENU



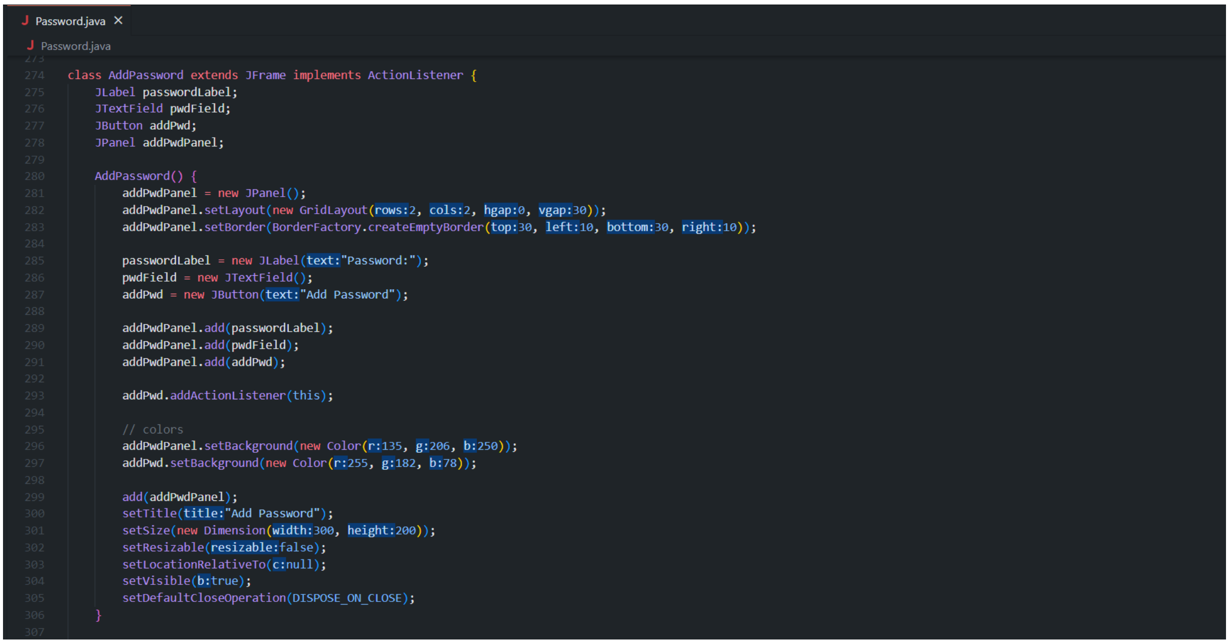
```java
class MainPage extends JFrame implements ActionListener {

    JButton listPasswords, generatePassword, addPassword, closeButton;
    JPanel mainPanel;

    MainPage() {
        mainPanel = new JPanel();
        mainPanel.setLayout(new GridLayout(rows:4, cols:1, hgap:0, vgap:30));
        mainPanel.setBorder(BorderFactory.createEmptyBorder(top:30, left:10, bottom:30, right:10));

        listPasswords = new JButton(text:"List Passwords");
        generatePassword = new JButton(text:"Generate Password");
        addPassword = new JButton(text:"Add New Password");
        closeButton = new JButton(text:"Close");

        mainPanel.add(listPasswords);
        mainPanel.add(generatePassword);
        mainPanel.add(addPassword);
        mainPanel.add(closeButton);

        // action listeners
        listPasswords.addActionListener(this);
        generatePassword.addActionListener(this);
        addPassword.addActionListener(this);
        closeButton.addActionListener(this);

        // colors
        mainPanel.setBackground(new Color(r:135, g:206, b:250));
        listPasswords.setBackground(new Color(r:255, g:182, b:78));
        generatePassword.setBackground(new Color(r:255, g:182, b:78));
        addPassword.setBackground(new Color(r:255, g:182, b:78));
        closeButton.setBackground(new Color(r:255, g:182, b:78));

        add(mainPanel);
        setTitle(title:"Home");
        setSize(new Dimension(width:300, height:400));
        setResizable(resizable:false);
        setLocationRelativeTo(c:null);
```

2.2 GENERATE PASSWORDS

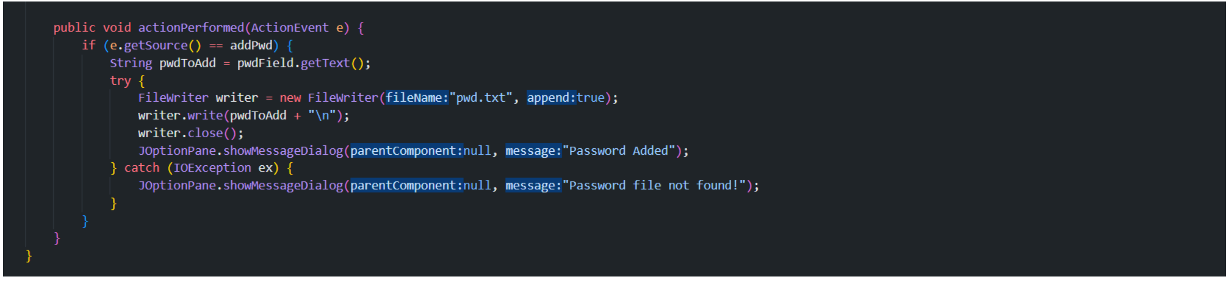


```java
                dispose();
            }
        }

        // password generator
        private char[] genPassword(int length) {
            String capitalCaseLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
            String lowerCaseLetters = "abcdefghijklmnopqrstuvwxyz";
            String specialCharacters = "!@#$";
            String numbers = "1234567890";
            String combinedChars = capitalCaseLetters + lowerCaseLetters + specialCharacters + numbers;
            Random random = new Random();
            char[] password = new char[length];

            password[0] = lowerCaseLetters.charAt(random.nextInt(lowerCaseLetters.length()));
            password[1] = capitalCaseLetters.charAt(random.nextInt(capitalCaseLetters.length()));
            password[2] = specialCharacters.charAt(random.nextInt(specialCharacters.length()));
            password[3] = numbers.charAt(random.nextInt(numbers.length()));

            for (int i = 4; i < length; i++) {
                password[i] = combinedChars.charAt(random.nextInt(combinedChars.length()));
            }
            return password;
        }

    }
```

2.3 ADD PASSWORDS

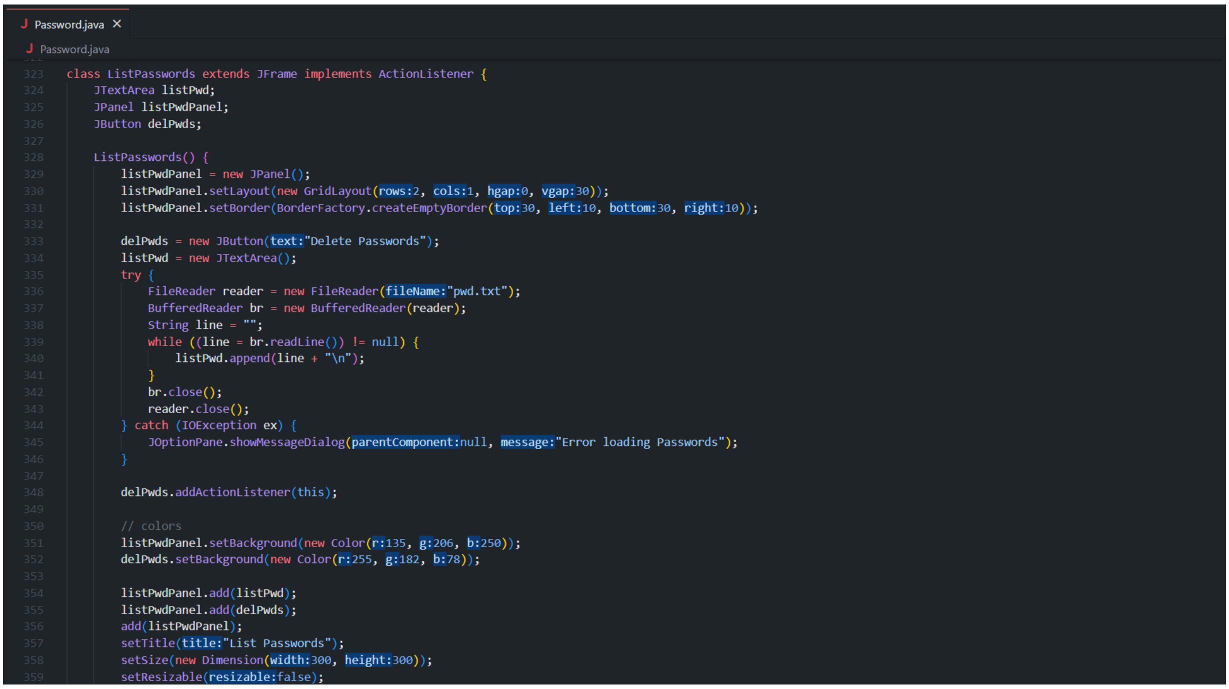


```java
class AddPassword extends JFrame implements ActionListener {
    JLabel passwordLabel;
    JTextField pwdField;
    JButton addPwd;
    JPanel addPwdPanel;

    AddPassword() {
        addPwdPanel = new JPanel();
        addPwdPanel.setLayout(new GridLayout(rows:2, cols:2, hgap:0, vgap:30));
        addPwdPanel.setBorder(BorderFactory.createEmptyBorder(top:30, left:10, bottom:30, right:10));

        passwordLabel = new JLabel(text:"Password:");
        pwdField = new JTextField();
        addPwd = new JButton(text:"Add Password");

        addPwdPanel.add(passwordLabel);
        addPwdPanel.add(pwdField);
        addPwdPanel.add(addPwd);

        addPwd.addActionListener(this);

        // colors
        addPwdPanel.setBackground(new Color(r:135, g:206, b:250));
        addPwd.setBackground(new Color(r:255, g:182, b:78));

        add(addPwdPanel);
        setTitle(title:"Add Password");
        setSize(new Dimension(width:300, height:200));
        setResizable(resizable:false);
        setLocationRelativeTo(c:null);
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
    }

```

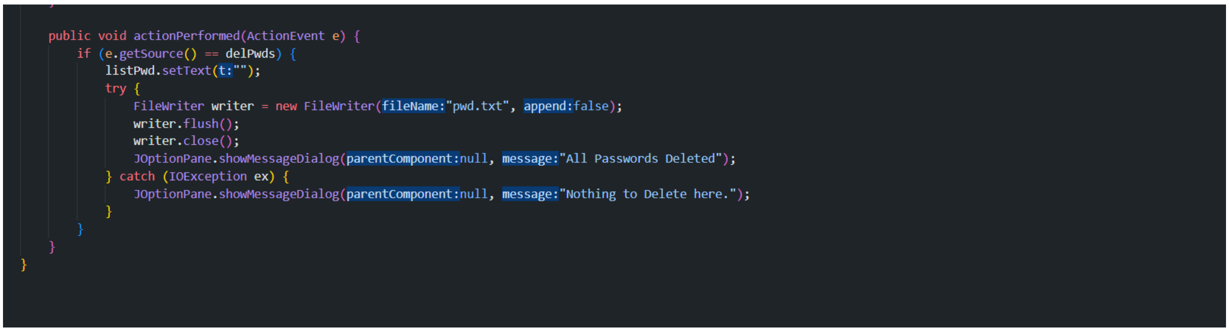


```java
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == addPwd) {
            String pwdToAdd = pwdField.getText();
            try {
                FileWriter writer = new FileWriter(fileName:"pwd.txt", append:true);
                writer.write(pwdToAdd + "\n");
                writer.close();
                JOptionPane.showMessageDialog(parentComponent:null, message:"Password Added");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Password file not found!");
            }
        }
    }
}
```

2.4 LIST PASSWORDS

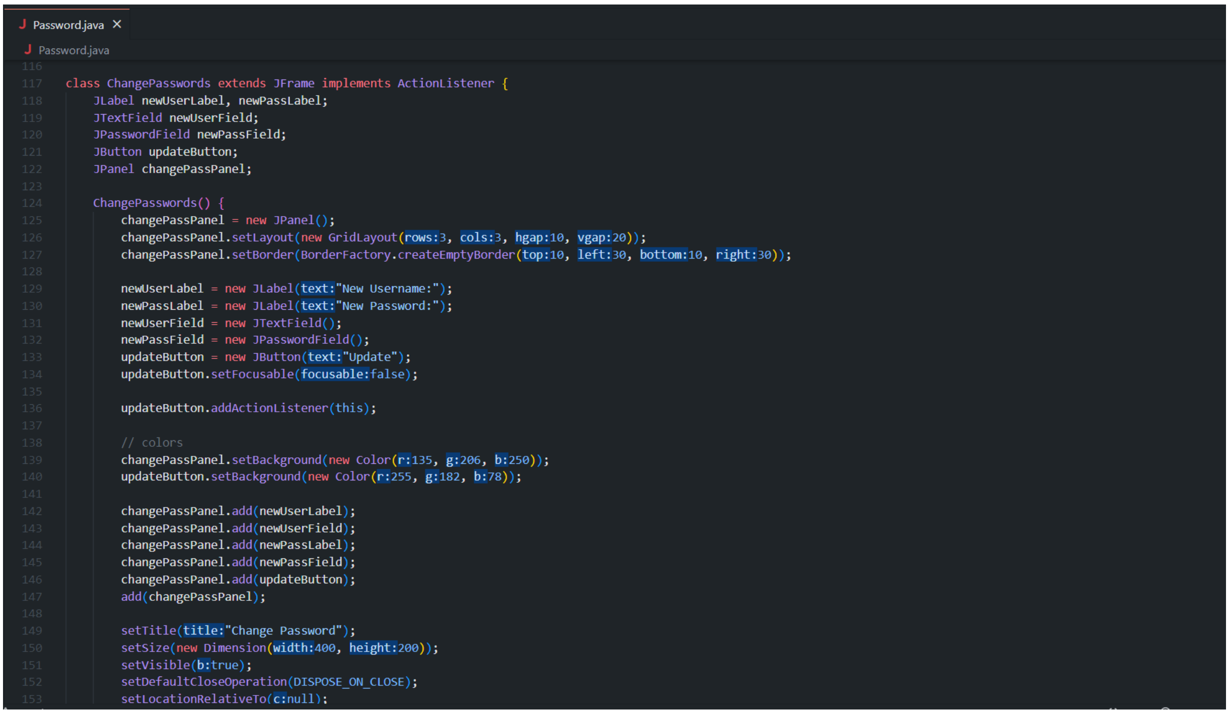


```java
class ListPasswords extends JFrame implements ActionListener {
    JTextArea listPwd;
    JPanel listPwdPanel;
    JButton delPwds;

    ListPasswords() {
        listPwdPanel = new JPanel();
        listPwdPanel.setLayout(new GridLayout(rows:2, cols:1, hgap:0, vgap:30));
        listPwdPanel.setBorder(BorderFactory.createEmptyBorder(top:30, left:10, bottom:30, right:10));

        delPwds = new JButton(text:"Delete Passwords");
        listPwd = new JTextArea();
        try {
            FileReader reader = new FileReader(fileName:"pwd.txt");
            BufferedReader br = new BufferedReader(reader);
            String line = "";
            while ((line = br.readLine()) != null) {
                listPwd.append(line + "\n");
            }
            br.close();
            reader.close();
        } catch (IOException ex) {
            JOptionPane.showMessageDialog(parentComponent:null, message:"Error loading Passwords");
        }

        delPwds.addActionListener(this);

        // colors
        listPwdPanel.setBackground(new Color(r:135, g:206, b:250));
        delPwds.setBackground(new Color(r:255, g:182, b:78));

        listPwdPanel.add(listPwd);
        listPwdPanel.add(delPwds);
        add(listPwdPanel);
        setTitle(title:"List Passwords");
        setSize(new Dimension(width:300, height:300));
        setResizable(resizable:false);
```

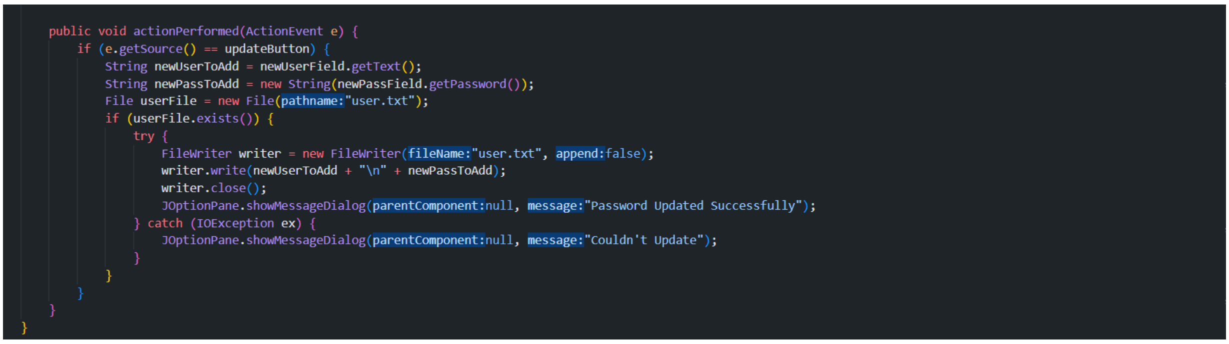


```java
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == delPwds) {
            listPwd.setText(t:"");
            try {
                FileWriter writer = new FileWriter(fileName:"pwd.txt", append:false);
                writer.flush();
                writer.close();
                JOptionPane.showMessageDialog(parentComponent:null, message:"All Passwords Deleted");
            } catch (IOException ex) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Nothing to Delete here.");
            }
        }
    }
}
```

2.5 CHANGE PASSWORDS

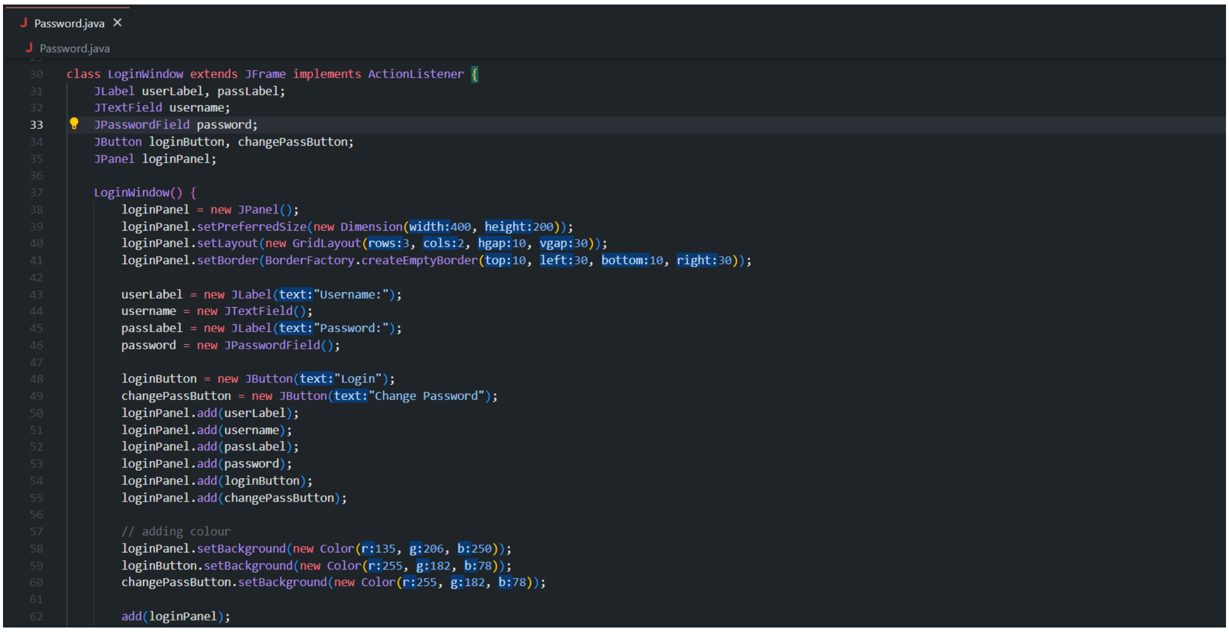


```

class ChangePasswords extends JFrame implements ActionListener {
    JLabel newUserLabel, newPassLabel;
    JTextField newUserField;
    JPasswordField newPassField;
    JButton updateButton;
    JPanel changePassPanel;

    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:250));
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
```

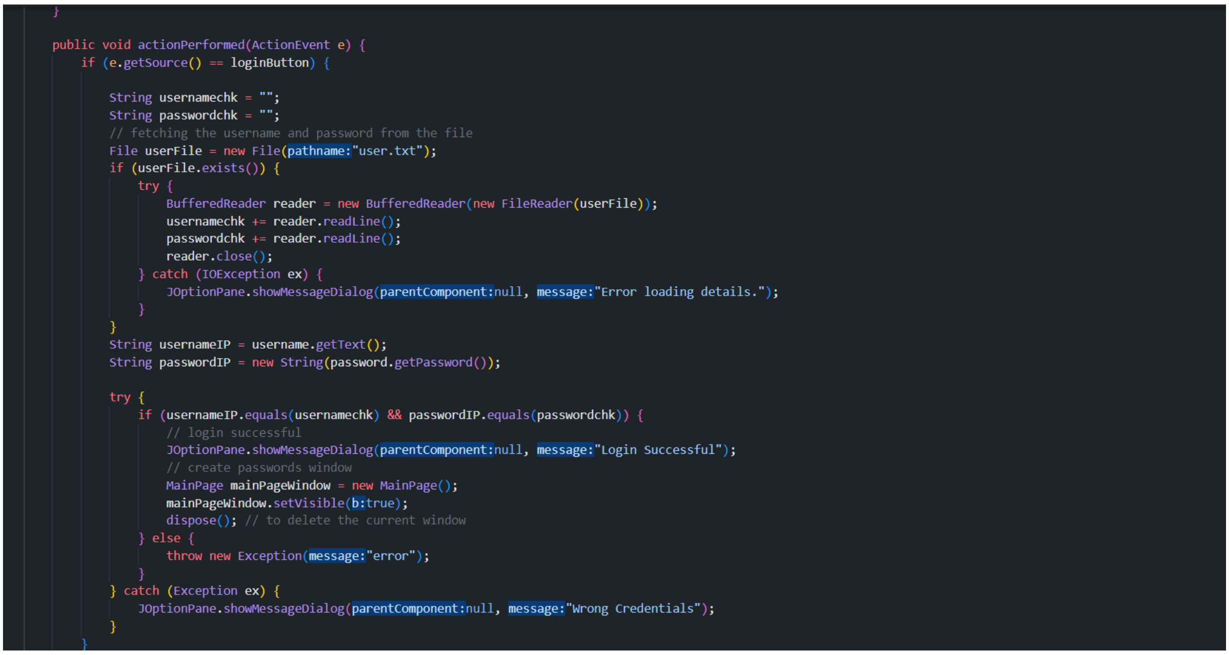


```
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == updateButton) {
            String newUserToAdd = newUserField.getText();
            String newPassToAdd = new String(newPassField.getPassword());
            File userFile = new File(pathname:"user.txt");
            if (userFile.exists()) {
                try {
                    FileWriter writer = new FileWriter(fileName:"user.txt", append:false);
                    writer.write(newUserToAdd + "\n" + newPassToAdd);
                    writer.close();
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Password Updated Successfully");
                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Couldn't Update");
                }
            }
        }
    }
}
```

2.6 LOGIN WINDOW



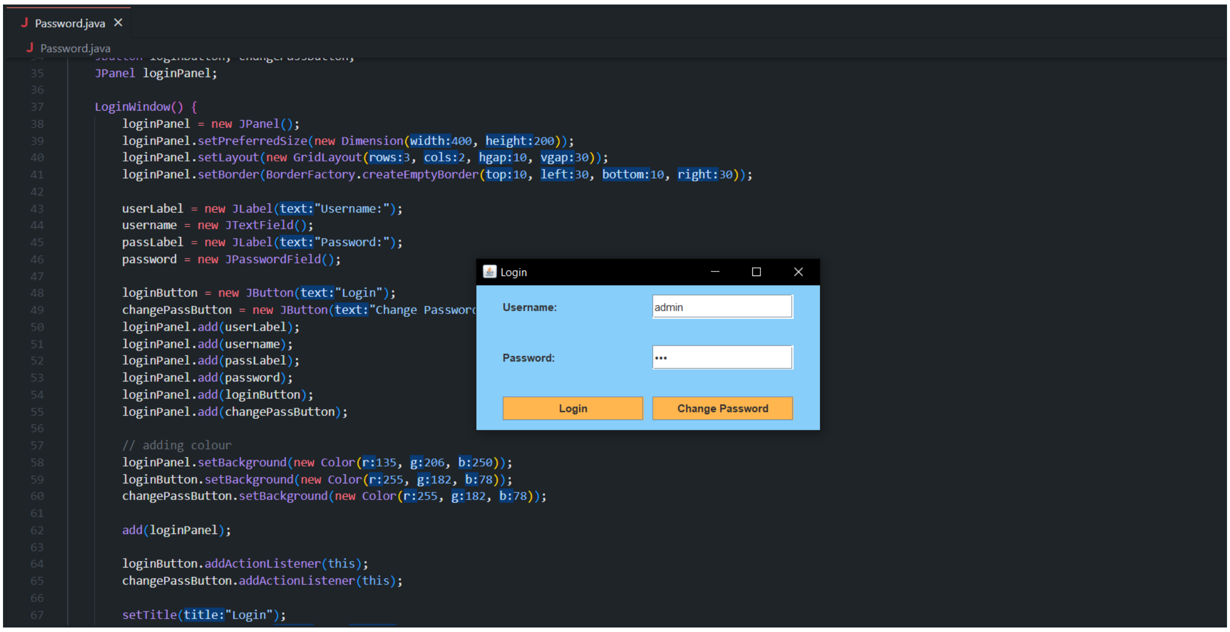
```java
class LoginWindow extends JFrame implements ActionListener {
    JLabel userLabel, passLabel;
    JTextField username;
    JPasswordField password;
    JButton loginButton, changePassButton;
    JPanel loginPanel;

    LoginWindow() {
        loginPanel = new JPanel();
        loginPanel.setPreferredSize(new Dimension(width:400, height:200));
        loginPanel.setLayout(new GridLayout(rows:3, cols:2, hgap:10, vgap:30));
        loginPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        userLabel = new JLabel(text:"Username:");
        username = new JTextField();
        passLabel = new JLabel(text:"Password:");
        password = new JPasswordField();

        loginButton = new JButton(text:"Login");
        changePassButton = new JButton(text:"Change Password");
        loginPanel.add(userLabel);
        loginPanel.add(username);
        loginPanel.add(passLabel);
        loginPanel.add(password);
        loginPanel.add(loginButton);
        loginPanel.add(changePassButton);

        // adding colour
        loginPanel.setBackground(new Color(r:135, g:206, b:250));
        loginButton.setBackground(new Color(r:255, g:182, b:78));
        changePassButton.setBackground(new Color(r:255, g:182, b:78));

        add(loginPanel);
```



```java
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == loginButton) {

            String usernamechk = "";
            String passwordchk = "";
            // fetching the username and password from the file
            File userFile = new File(pathname:"user.txt");
            if (userFile.exists()) {
                try {
                    BufferedReader reader = new BufferedReader(new FileReader(userFile));
                    usernamechk += reader.readLine();
                    passwordchk += reader.readLine();
                    reader.close();
                } catch (IOException ex) {
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Error loading details.");
                }
            }
            String usernameIP = username.getText();
            String passwordIP = new String(password.getPassword());

            try {
                if (usernameIP.equals(usernamechk) && passwordIP.equals(passwordchk)) {
                    // login successful
                    JOptionPane.showMessageDialog(parentComponent:null, message:"Login Successful");
                    // create passwords window
                    MainPage mainPageWindow = new MainPage();
                    mainPageWindow.setVisible(b:true);
                    dispose(); // to delete the current window
                } else {
                    throw new Exception(message:"error");
                }
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Wrong Credentials");
            }
        }
```

3.MODULE 2 (OUTPUT)

3.1 LOGIN WINDOW



Password.java

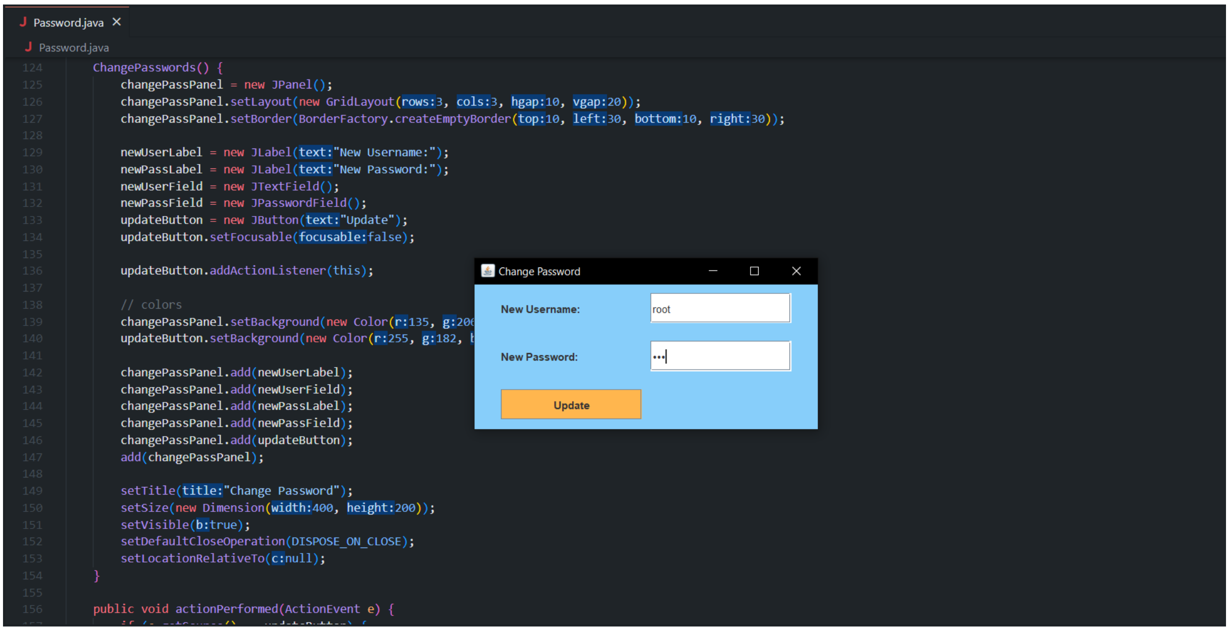
```
    JButton loginButton, changePassButton;
    JPanel loginPanel;

    LoginWindow() {
        loginPanel = new JPanel();
        loginPanel.setPreferredSize(new Dimension(width:400, height:200));
        loginPanel.setLayout(new GridLayout(rows:3, cols:2, hgap:10, vgap:30));
        loginPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        userLabel = new JLabel(text:"Username:");
        username = new JTextField();
        passLabel = new JLabel(text:"Password:");
        password = new JPasswordField();

        loginButton = new JButton(text:"Login");
        changePassButton = new JButton(text:"Change Passwor
        loginPanel.add(userLabel);
        loginPanel.add(username);
        loginPanel.add(passLabel);
        loginPanel.add(password);
        loginPanel.add(loginButton);
        loginPanel.add(changePassButton);

        // adding colour
        loginPanel.setBackground(new Color(r:135, g:206, b:250));
        loginButton.setBackground(new Color(r:255, g:182, b:78));
        changePassButton.setBackground(new Color(r:255, g:182, b:78));

        add(loginPanel);

        loginButton.addActionListener(this);
        changePassButton.addActionListener(this);

        setTitle(title:"Login");
```

Login

Username: admin

Password: •••

3.2 PASSWORD UPDATE



Password.java

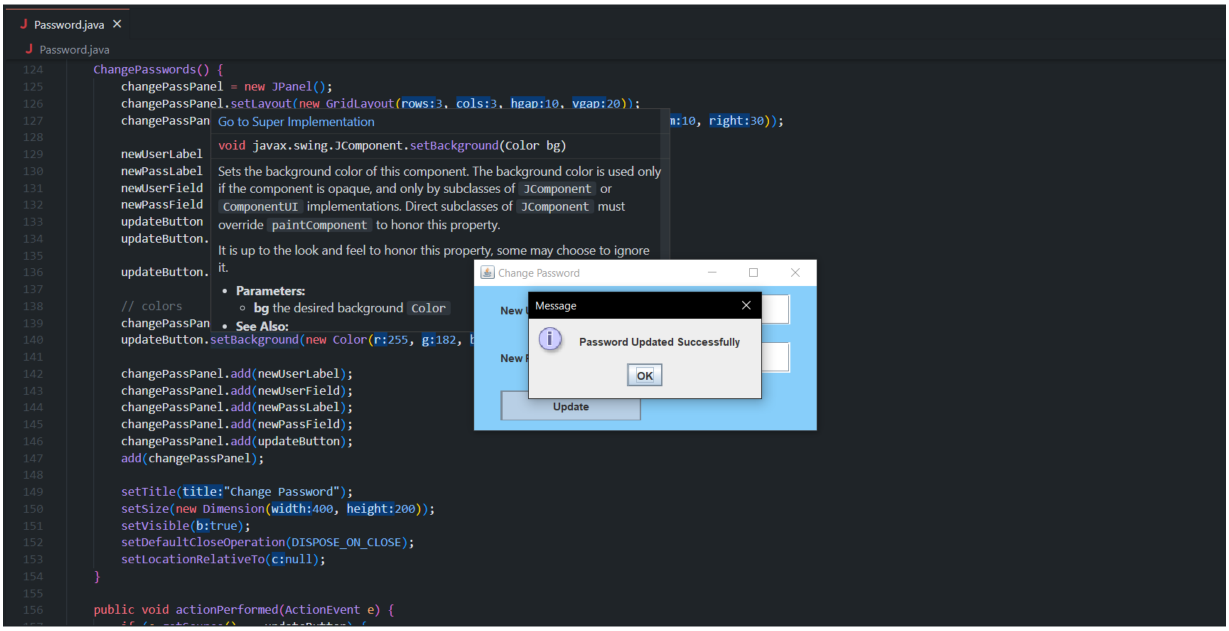
```
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:20
        updateButton.setBackground(new Color(r:255, g:182, b

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == updateButton) {
```

Change Password

New Username: root

New Password: •••

3.3 PASSWORD UPDATE SUCCESSFUL



Password.java ✕

Password.java

```
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPan                                                          n:10, right:30));

        newUserLabel
        newPassLabel
        newUserField
        newPassField
        updateButton
        updateButton.

        updateButton.

        // colors
        changePassPan
        updateButton.setBackground(new Color(r:255, g:182, b

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```

Go to Super Implementation

`void javax.swing.JComponent.setBackground(Color bg)`

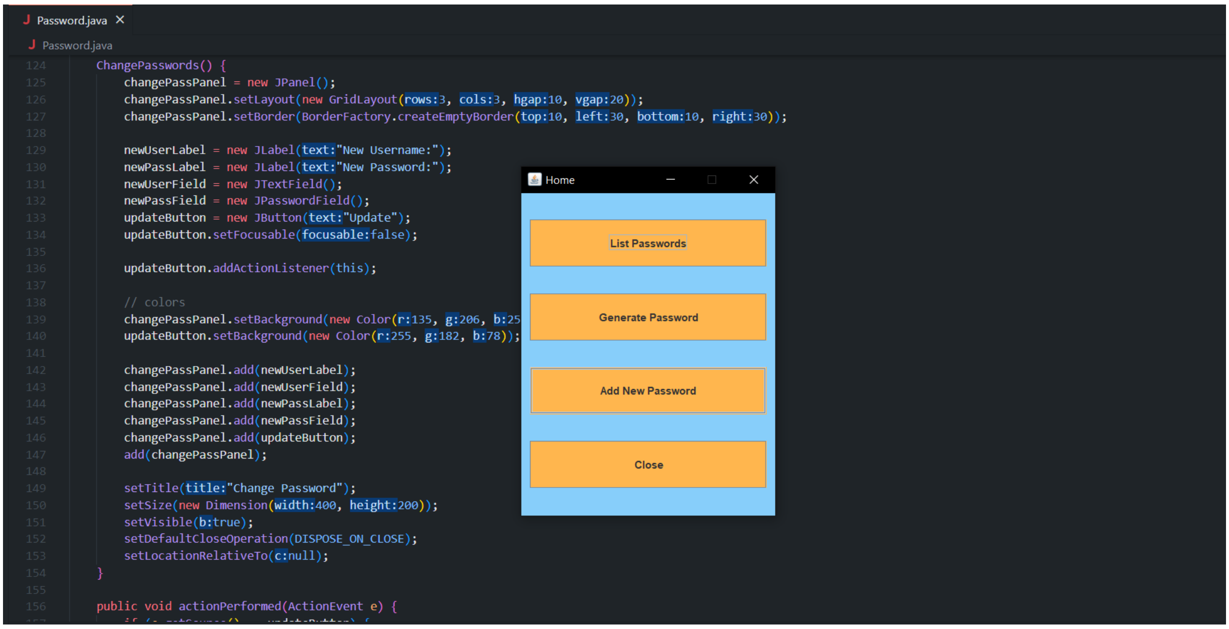
Sets the background color of this component. The background color is used only if the component is opaque, and only by subclasses of `JComponent` or `ComponentUI` implementations. Direct subclasses of `JComponent` must override `paintComponent` to honor this property.

It is up to the look and feel to honor this property, some may choose to ignore it.

- **Parameters:**
  - **bg** the desired background `Color`
- **See Also:**

Password Updated Successfully

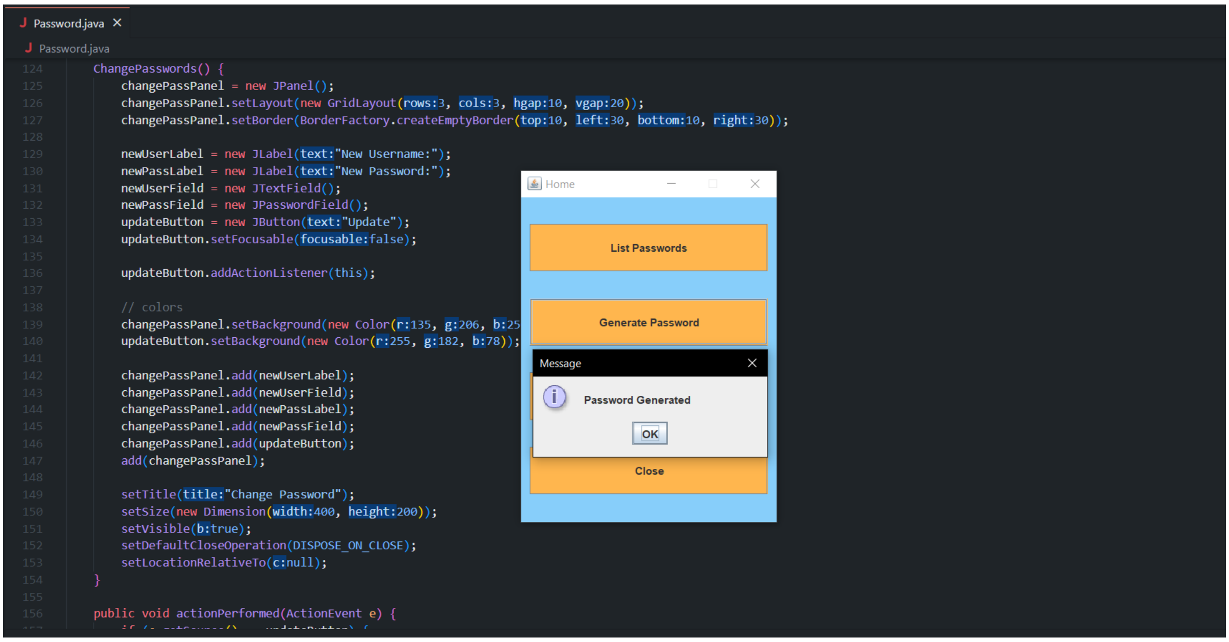
3.4 MAIN PAGE SIMULATION



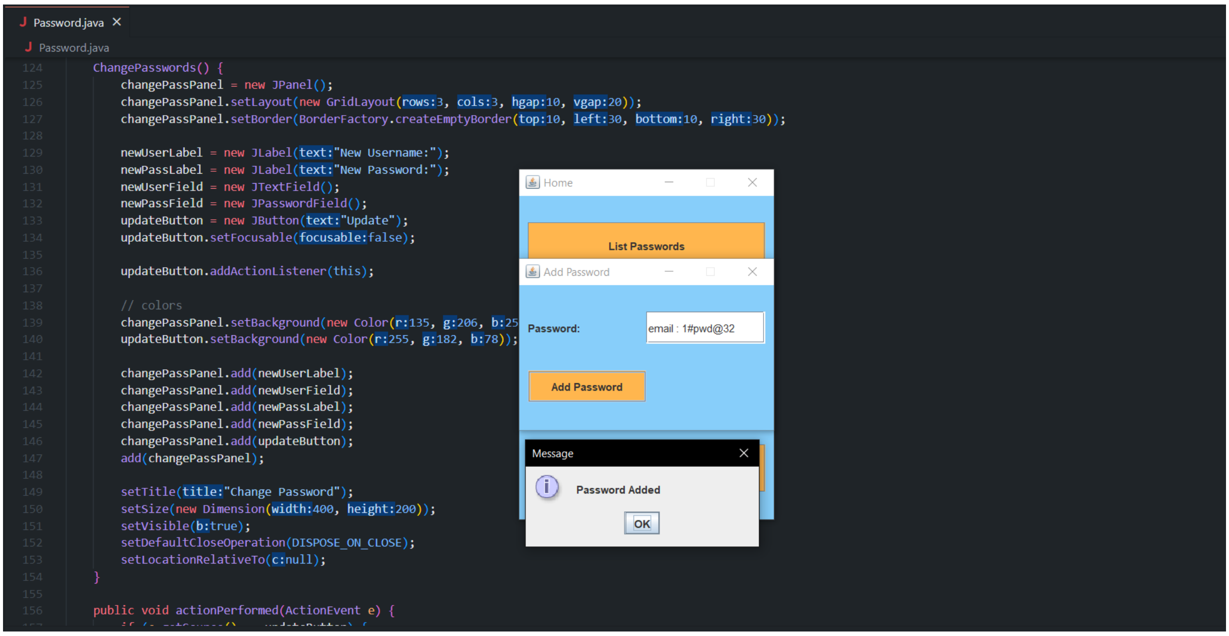
Password.java ✕

Password.java

```
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:25
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```

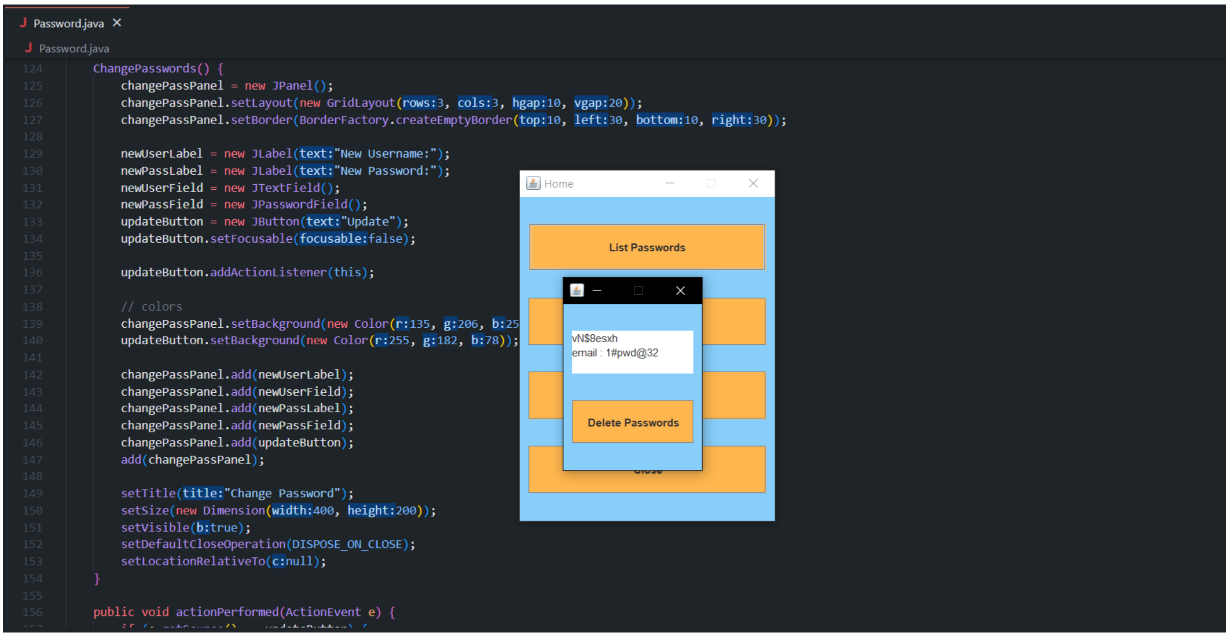


```java
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:25
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```



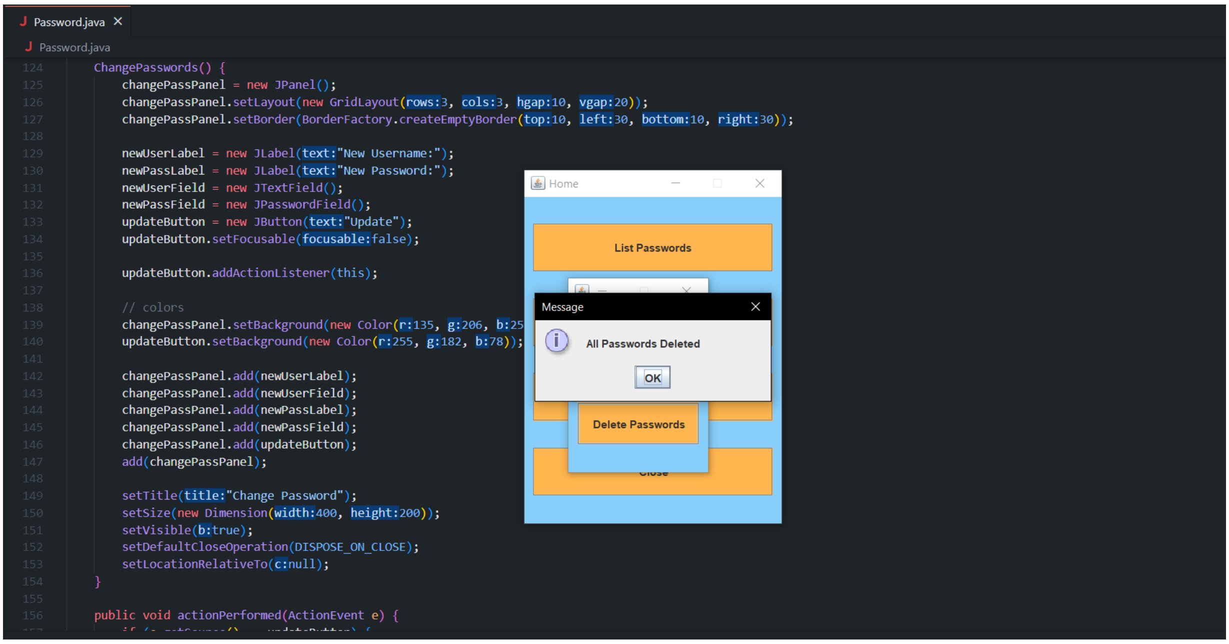
```java
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:25
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```

Password: email : 1#pwd@32



```java
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:25
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```

vN$8esxh
email : 1#pwd@32



```java
    ChangePasswords() {
        changePassPanel = new JPanel();
        changePassPanel.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:20));
        changePassPanel.setBorder(BorderFactory.createEmptyBorder(top:10, left:30, bottom:10, right:30));

        newUserLabel = new JLabel(text:"New Username:");
        newPassLabel = new JLabel(text:"New Password:");
        newUserField = new JTextField();
        newPassField = new JPasswordField();
        updateButton = new JButton(text:"Update");
        updateButton.setFocusable(focusable:false);

        updateButton.addActionListener(this);

        // colors
        changePassPanel.setBackground(new Color(r:135, g:206, b:25
        updateButton.setBackground(new Color(r:255, g:182, b:78));

        changePassPanel.add(newUserLabel);
        changePassPanel.add(newUserField);
        changePassPanel.add(newPassLabel);
        changePassPanel.add(newPassField);
        changePassPanel.add(updateButton);
        add(changePassPanel);

        setTitle(title:"Change Password");
        setSize(new Dimension(width:400, height:200));
        setVisible(b:true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        setLocationRelativeTo(c:null);
    }

    public void actionPerformed(ActionEvent e) {
```

Message

All Passwords Deleted

4. CONCLUSION

1. The project provides a simple graphical user interface to store and manage passwords.

2. Users can change their login credentials and access the stored passwords.

3. The project offers three main functionalities for password management. Listing passwords, generating passwords, and adding new passwords.

4. The project uses file handling methods to validate login credentials and read and write user passwords.

5. The project was made interactive and simple using java swing and AWT.

6. The project offers an easy-to-use solution for password management making it a useful tool for individuals.

5. FUTURE SCOPE

1. Implementing database integration to store and retrieve user data and passwords. Adding encryption and decryption methods to make the application more secure.

2. Adding a password strength checker to help users set strong passwords.

3. Adding password expiration policies.

4. Implementing multi-factor authentication for enhanced security.

6. REFERENCE

1.YouTube

2.JAVA Oracle for JFrames.