**Outline**

The program has 2 entities

1. User
2. Admin

User:

User will be of two types:

1. Free User
2. Premium User

Free User can save upto 5 passwords only

Premium users can save infinite amount of passwords

Users must sign up

Admin:

Admin have the privilege to view the total number of users in the system

Admins will be given a password from which they have to sign in

Admins can delete user from the system

Program:

In a loop

Prompt user the message “Are you a regular user or an admin if regular user press 1 if admin press 2 press 3 to exit”

If 3 then exit the program

Check the users input if anything else is entered other than 1 or 2 or 3 give message “Please enter appropriate input”

If user selects admin give message “Please enter the password”

Check the password against admin.txt file if it does not match say “Password is incorrect”

In a loop

If password is correct then prompt “What do you want to do today if you want to view users press 1 if you want to delete users press 2 press 3 to log out”

If 3 then log out

If admin selects 1 save details from user.txt file in an array

Then show them the details from array

If admin selects 2 prompt “Enter the username of user you want to delete”

Delete the row of the username admin entered

Delete the username.txt file

If user selects 1

Give message “Are you a new or an existing user if existing press 1 else press 2 to signup”

In a loop

If user press 1 prompt” enter email and password of their account”

Check against user.txt file if email or password is incorrect say “Please enter correct email and password”

If correct give prompt “Press 1 if you want to see websites and passwords you save if you want to enter new password press 2 press 3 to logout”

If 3 then log out

If 1 is pressed show them the passwords from their username.txt file

If 2 is pressed prompt “Enter website name and password”

Save it in username.txt file

If user press 2 “Prompt enter username email and password to sign up ”

Verify the email address using email address verifier method

If email invalid prompt “Enter a valid email address”

Else create a txt file with the username.txt and prompt “Account created successfully “

Prompt “Press 2 to become a premium user 1 to remain free user”

If pressed 2 in users.txt file write the status of user as “Premium”

If pressed 1 in users.txt file write the status of user as “Free”

Loop

Prompt “Press 1 if you want to see websites and passwords you save if you want to enter new password press 2 press 3 to logout”

If 3 then log out

If admin selects 1 save details from usernamme.txt file in an array

If 2 is pressed check if they have more than 5 passwords save in their file

If more than 5 Prompt “Free users are not allowed more than 5 passwords”

**Logic**

User Class

Constructor:

Sets the user's name and password.

exists:

1. Open the file with the user's name and extension `.txt`.

2. If the file is open:

\* Get the first line of the file.

\* If the first line is the user's password:

\* Close the file.

\* Return `true`.

\* Close the file.

3. Return `false`.

GetName()

Returns users name

GetPassword()

Returns users password

signup:

1. Create a file with the user's name and extension `.txt`.

2. Write the user's password to the file.

3. Open the `users.txt` file in append mode.

4. Write the user's name, email, and status to the file.

5. Close the file.

write\_passwords:

1. Create a file with the user's name and extension `.txt`.

2. Open the file in append mode.

3. Write the website and password to the file, separated by a space.

4. Close the file.change\_password:

1. Open the file with the user's name and extension `.txt`.

2. Create a temporary file.

3. Iterate over the lines in the file:

\* If the line contains the website name:

\* Replace the password on the line with the new password.

\* Write the line to the temporary file.

\* Otherwise, write the line to the temporary file unchanged.

4. Close the file and the temporary file.

5. Delete the original file.

6. Rename the temporary file to the original file name.

7. If the password was changed successfully, print a message to the user.

8. Otherwise, print a message to the user asking them to check the website name.

view\_passwords:

1. Open the file with the user's name and extension `.txt`.

2. Create an array of strings to store the passwords.

3. Iterate over the lines in the file:

\* Add the line to the array of strings.

4. Close the file.

5. Iterate over the array of strings, starting at index 1:

\* Print the string to the console.

6. Print the number of passwords to the console..

Admin Class

remove\_user:

1. Open the file with the given path.

2. Create a temporary file.

3. Iterate over the lines in the file:

\* If the line does not contain the given username:

\* Write the line to the temporary file.

\* Otherwise:

\* Set the `found` flag to `true`.

4. Close the file and the temporary file.

5. Delete the original file.

6. Rename the temporary file to the original file name.

7. If the user was found, print a message to the console.

8. Otherwise, print a message to the console asking the user to check the username.

view\_users:

1. Open the file with the given name.

2. Create an array of strings to store the usernames.

3. Iterate over the lines in the file:

\* Add the line to the array of strings.

4. Close the file.

5. Iterate over the array of strings:

\* Print the string to the console.

Main function:

1. Declare a variable `determine` of type `int`.

2. Prompt the user to enter 1 for regular user, 2 for admin, or 3 to exit.

3. If the user enters 1, do the following:

\* Prompt the user to enter 1 for an existing user or 2 for a new user.

\* If the user enters 1, do the following:

\* Prompt the user to enter their username and password.

\* Check if the username and password exist in the `users.txt` file.

\* If they exist, do the following:

\* Welcome the user to the password manager and display the following menu:

\* View passwords

\* Write down new passwords

\* Change passwords

\* Logout

\* Repeat the menu until the user chooses to logout.

\* If the username and password do not exist, do the following:

\* Display an error message and ask the user to try again.

\* If the user enters 2, do the following:

\* Prompt the user to enter their username, email, and password.

\* Check if the username already exists in the `users.txt` file.

\* If the username does not exist, do the following:

\* Write the username, email, and password to the `users.txt` file.

\* Display a success message.

\* If the username does exist, do the following:

\* Display an error message and ask the user to try again.

4. If the user enters 2, do the following:

\* Prompt the user to enter the admin password.

\* Check if the admin password is correct.

\* If the admin password is correct, do the following:

\* Display the following menu:

\* View users

\* Delete users

\* Logout

\* Repeat the menu until the user chooses to logout.

\* If the admin password is incorrect, do the following:

\* Display an error message and ask the user to try again.

5. If the user enters 3, do the following:

\* Exit the program.