

Paul Fitch

IDE: VS Code 1.79

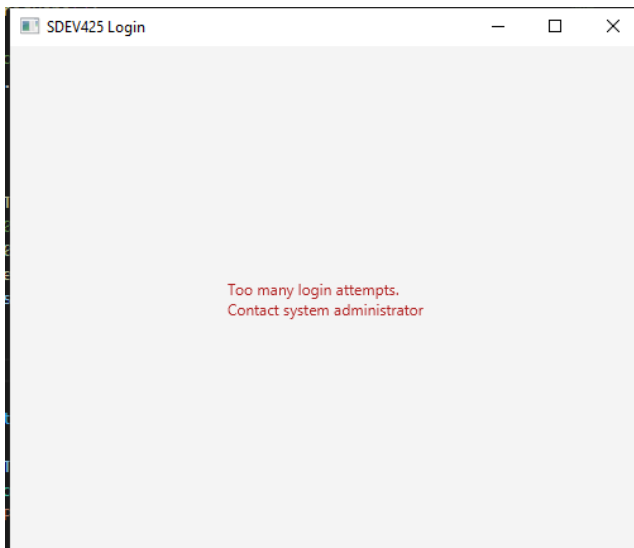
Relevant extensions:

- Debugger for java
- Extension pack for java
- Language support for java
- Project manager for java
- Test runner for java
- Javafx support

MIST low-impact security controls:

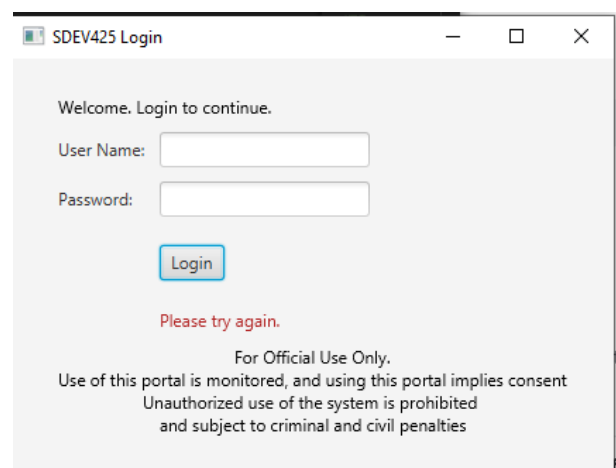
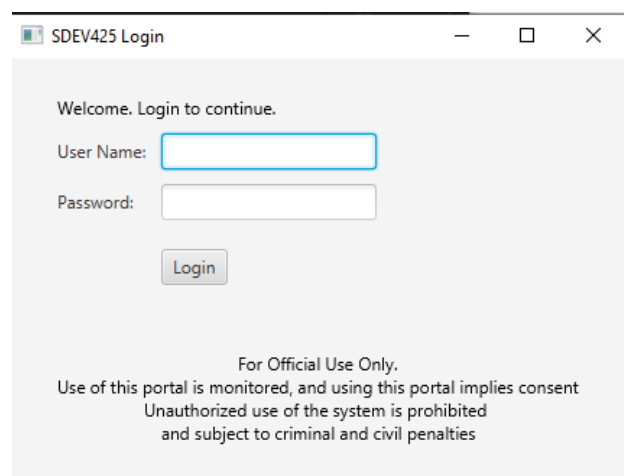
AC - 7 Unsuccessful logon attempts:

Implemented limit on number unsuccessful login attempts. The application will now lock (at the application level) upon 3 unsuccessful login attempts



AC - 8 System use message:

Implemented a message on the login grid visible below the input boxes and login button. It is in a position that does not interfere with "Please try again" warning text



AU - 3 Content of Audit Records:

Implemented audit records that record type of event (i.e. login fail, login success), IP address of computer connected to application, and timestamp of when the event occurred. This feature also incorporates the AU-3(3) enhancement, limiting the amount of PII stored in the audit records.

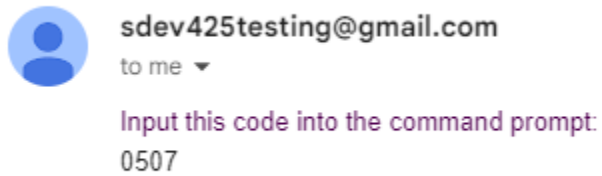
```
1 Login fail for username: sdevadmin
2 From IP Address: MSI/100.127.255.253
3 at: 11-06-2023 13:54
4
5 Login fail for username: sdevadmin
6 From IP Address: MSI/100.127.255.253
7 at: 11-06-2023 13:54
8
9 Login fail from IP Address: MSI/100.127.255.253
10 at: 11-06-2023 13:54
11 | Account with username sdevadmin locked
12
13 Login fail for username: sdevadmin
14 From IP Address: MSI/100.127.255.253
15 at: 11-06-2023 13:56
16
17 Login success for username: user
18 From IP Address: MSI/100.127.255.253
19 at: 11-06-2023 13:56
```

AU - 8 Timestamps:

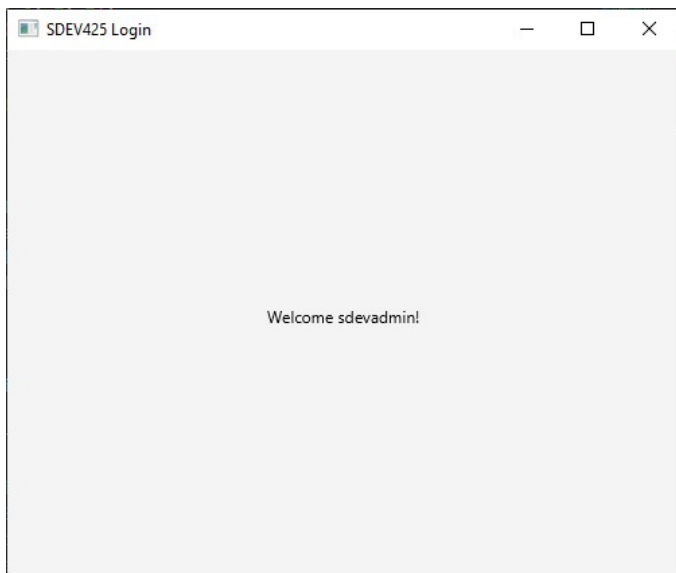
Implemented a timestamp as part of every audit record.

IA 2(1) Identification and authorization of organizational users:

Implemented privilege levels, and code to discern privilege levels. Implemented 2 factor email authentication for users with higher privilege levels. Users with low privilege do not require 2 step authentication.

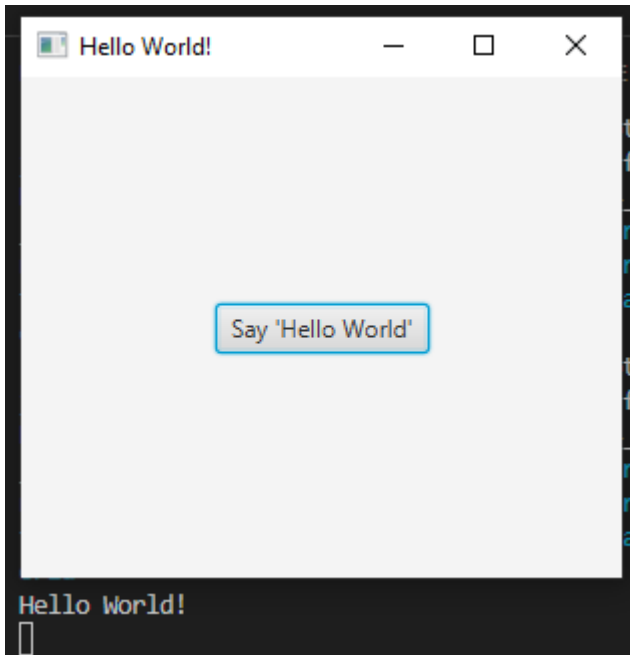


```
Enter email used for two factor authentication
plftch9@gmail.com
Enter code from email:
0507
█
```



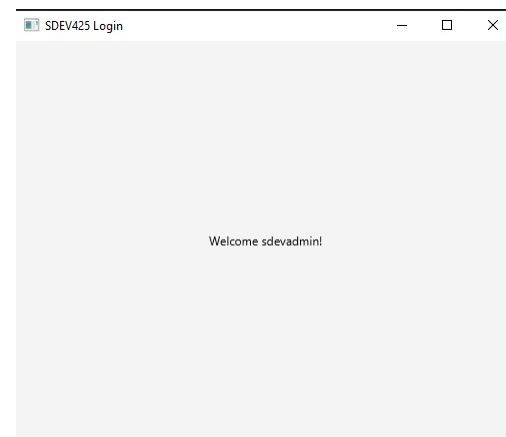
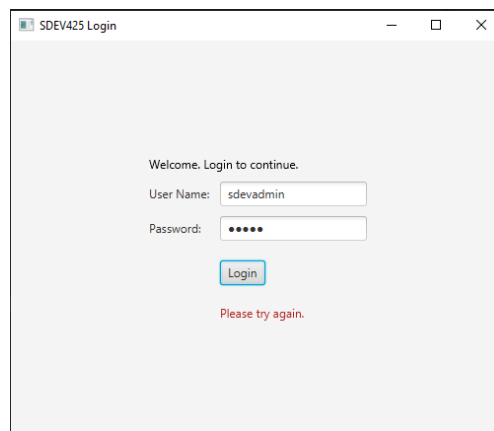
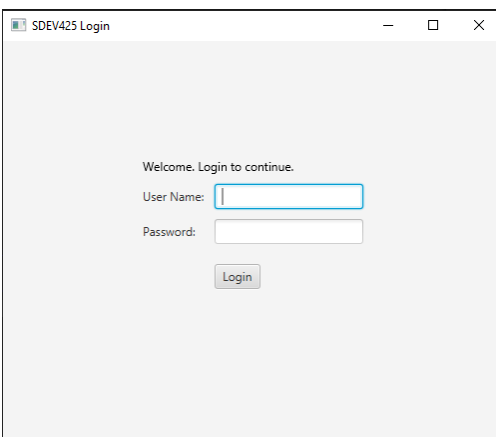
Sample JavaFX application Hello World:

The java fx hello, world described in the assignment is a netbeans specific type of template that is started when creating a javafx project in the netbeans project manager. Since I am using VS Code, I do not have the exact template in the assignment description, however I have made a new one:



Sample JavaFx application SDEV425_2:

Unmodified screen captures:



Overview:

The application uses the javafx library to construct and show a GUI used as a login portal, and a simple authentication function using hard coded credentials. Components of this GUI (Grid frame, 2 text boxes, a button, and several labels) are made and assigned in the "public void start()" method.

Once the components are set, a handler is used to read user input, though the defined handler only reads click inputs. This handler, on click, passed the text supplied to the text boxes to the authenticate method, which checks inputs against hard coded credentials. Upon authenticate returning true, the original grid frame is set to not visible, and a second frame welcomes the user. Upon authenticate returning false, a message appears on the original grid frame prompting the user to try logging in again.