Bryan Gaskins
05/26/22
Caltech CTME PGP: Full Stack Web Dev
Phase 3: Assignment #3: Handling User Authentication / Description and Process
GitHub: https://github.com/bgaskins/handling_user_authentication

Project objective:
Set up a standalone project to do unit testing of the user authentication class which is used in the main web application. The objective is to create a JUnit class that will test all aspects of the authentication class.

Background of the problem statement:
As a part of developing an e-commerce web application, a test-suite is being created to do unit testing of all backend components in the web application. This project will test the user authentication class. This project will be a standalone Java application, since Junit does not directly test servlets or web pages. We are only testing the classes that have the business logic.

You must use the following:
- Eclipse as the IDE
- Apache Tomcat as the web server
- Junit 5

Following requirements should be met:
- Create a standalone Java application using Maven
- Create an authentication class that has all the methods related to user authentication
- Create a JUnit test class to create unit tests for the authentication class
- Run the test class directly as a JUnit and check if all the tests pass
- Document the step-by-step process involved in completing this task

Process (step-by-step):
1. Create a standalone Java application using Maven
2. Create Authentication class with User Authentication methods: Login, getEmail, getUsername, Logout
3. Create JUnit test class to create unit tests for Authentication class
4. Run the test class directly as a JUnit and check if all the tests pass
5. Upload to GitHub

Authentication class:

```java
package com.HandlingUserAuth;

import java.util.HashSet;

public class Authentication {

    public static Set<UserEntity> userList = new HashSet<>();
    private UserEntity currentSessionUser = null;

    public Boolean login(String userName, String password) {

        AtomicBoolean userExists = new AtomicBoolean(false);
        userList.stream().filter(x -> x.getUserName().equals(userName) && x.getPassword().equals(password))
                .findFirst()
                .ifPresent(x -> {
                    userExists.set(true);
                    currentSessionUser = x;
                });
        return userExists.get();

    }

    public String getEmail() {
        if(currentSessionUser !=null) {
            return currentSessionUser.getEmail();
        }
        return  null;
    }

    public String getUserName() {
        if(currentSessionUser !=null) {
            return currentSessionUser.getUserName();
        }
        return  null;
    }

    public void logout() {
        currentSessionUser = null;
    }
}

```

JUnit test class with passed JUnit test:

```java
public class AuthenticationTest {
    @BeforeEach
    public void setup() {
        UserEntity user1 = new UserEntity("Carl", "pass", "carl@gmail.com");
        UserEntity user2 = new UserEntity("Mateo", "pass", "mateo@gmail.com");
        UserEntity user3 = new UserEntity("Nancy", "pass", "nancy@gmail.com");
        Authentication.userList.add(user1);
        Authentication.userList.add(user2);
        Authentication.userList.add(user3);
    }

    @Test
    public void testLogin() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("Carl", "pass"));
    }
    @Test
    public void testWrongUserLogin() {
        Authentication authentication = new Authentication();
        assertEquals(false, authentication.login("Tyler", "pass"));
    }
    @Test
    public void testEmailAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("Carl", "pass"));
        assertEquals("carl@gmail.com", authentication.getEmail());
    }
    @Test
    public void testUserNameAssert() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("Carl", "pass"));
        assertEquals("Carl", authentication.getUserName());
    }

    @Test
    public void testLogout() {
        Authentication authentication = new Authentication();
        assertEquals(true, authentication.login("Carl", "pass"));
        authentication.logout();
        assertEquals(null, authentication.getEmail());
    }
}
```