Reflection

Embarking on the intricate journey of creating a security-centric program for this assignment was akin to navigating the multifaceted world of cybersecurity. The endeavor was neither linear nor predictable. While I entered this project equipped with the theoretical principles from lectures and labs, the unique challenges that surfaced demanded a deeper dive into additional resources, leading to a synthesis of formal and informal learning.

Learning from Extra Resources:

The ceaselessly evolving domain of security development necessitates constant learning. While classroom teachings provided a solid foundation, the real-world application often requires tapping into the vast reservoir of knowledge online. Websites like StackOverflow became crucial touchpoints, illuminating complex aspects of programming. My exploration of cybersecurity blogs, online forums, and interactive platforms was equally rewarding, offering contemporary insights, solutions, and emerging threats in the field. These resources not only supplemented my academic learning but also introduced me to a community-driven approach to problem-solving.

Relating to Labs and Lectures:

Every academic session has been a cornerstone in shaping the program. The "Least Privilege" principle, for instance, transitioned from being a topic of academic discussion to a fundamental aspect of the program's security. By segregating users based on roles, the program fortified itself against potential unauthorized breaches.

The emphasis on data integrity and confidentiality in our curriculum was profoundly resonant. Real-world instances of security lapses discussed in lectures served as cautionary tales. This underscored the decision to integrate a two-factor authentication system, aiming to bolster the program's defense, especially when there's a risk of initial credentials being compromised.

Moreover, labs allowed me to experiment with different tools and techniques, making me appreciate the intricacies of security measures in a hands-on manner. The simulations and hands-on lab exercises were invaluable in demonstrating the practical implications of theoretical concepts.

Limitations and Proposed Solutions:

Scalability: Relying on in-memory structures like HashMap's was a conscious choice for immediate benefits but poses long-term scalability issues.

Solution: Transitioning to robust databases like PostgreSQL or MongoDB ensures scalability and consistent performance, catering to increasing user demands.

Security of Data Transmission: Password hashing is a step forward, but data transmission paths have vulnerabilities.

Solution: Implementing SSL/TLS is non-negotiable. It guarantees that data remains encrypted in transit, deterring unauthorized interception.

Centralized Authentication System: A singular access point, while efficient, is also a potential vulnerability.

Solution: Mechanisms like OAuth or OpenID offer decentralized authentication, spreading the risk and enhancing system resilience.

Mailjet SMTP Credentials Exposure: Directly embedding credentials is a glaring vulnerability.

Solution: Utilizing tools like HashiCorp's Vault or environment variables ensures that sensitive information remains concealed, safeguarded from potential threats.

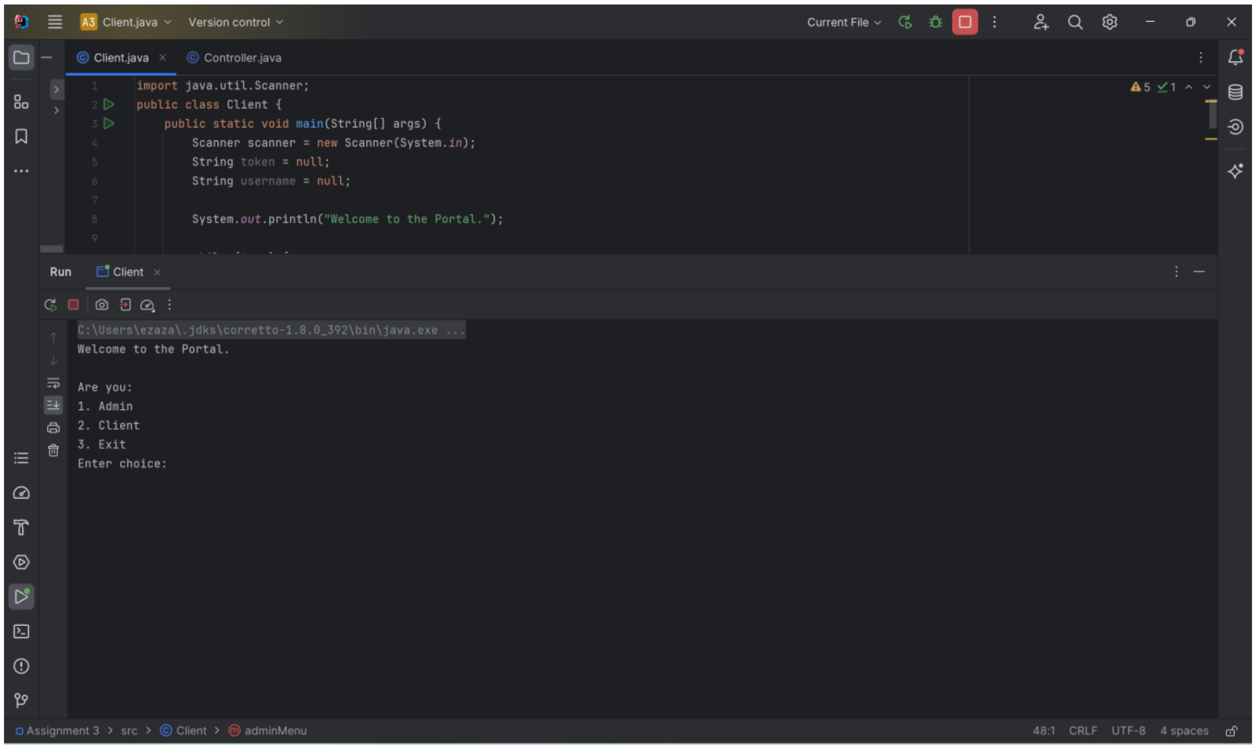
Conclusion:

Reflecting on this project, I realize it was more than a mere assignment; it was a profound learning journey. Beyond the lines of code, it was about understanding the nuances of cybersecurity, thinking critically, and always being proactive. The blend of academic principles with real-world application challenges has enriched my perspective. Recognizing the program's strengths and understanding its limitations has instilled in me the essence of continuous growth in cybersecurity. The project also underscored the importance of adaptability, as the dynamic nature of cybersecurity means new challenges can emerge at any time. Thus, staying updated and vigilant becomes crucial. I am left with an invigorated passion for delving deeper into this ever-evolving domain, with this assignment marking a pivotal milestone.

Bibliography:

StackOverflow. (2022). StackOverflow Community Discussion. StackExchange.

Screenshots of the Program execution:



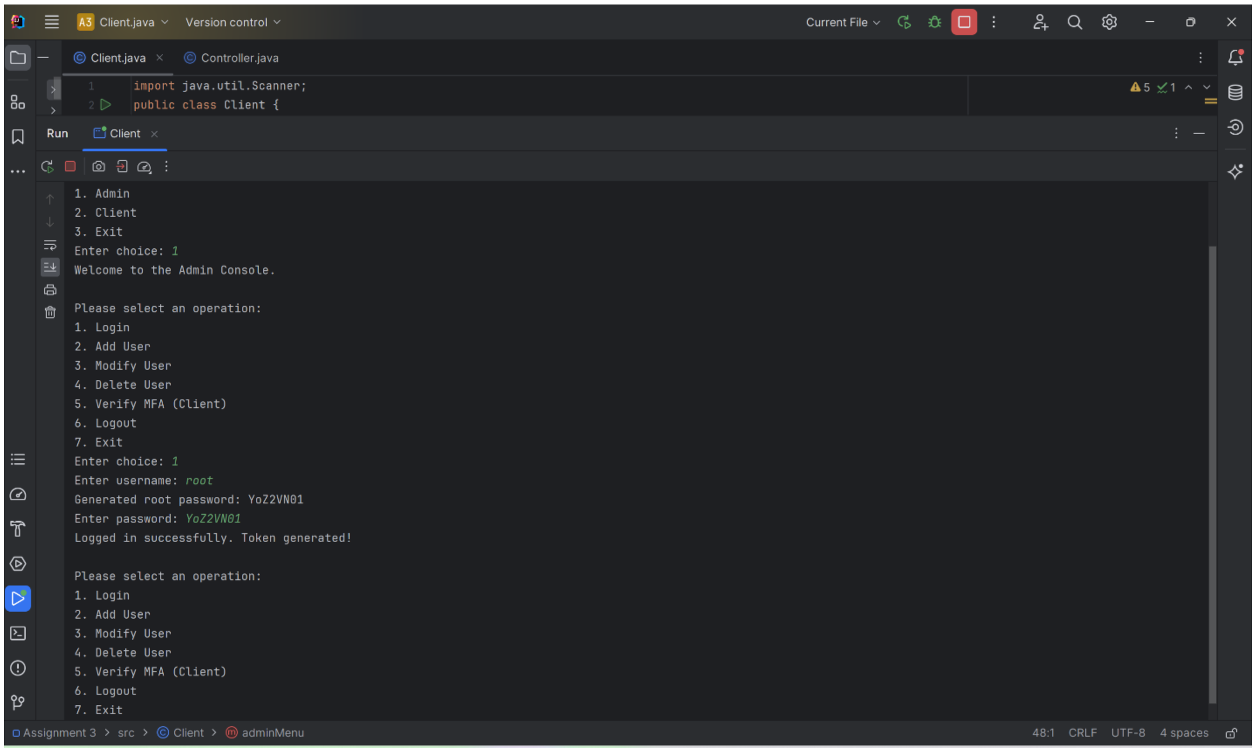
```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
C:\Users\ezaza\.jdks\corretto-1.8.0_392\bin\java.exe ...
Welcome to the Portal.

Are you:
1. Admin
2. Client
3. Exit
Enter choice:
```

- User will decide log in as a admin or client

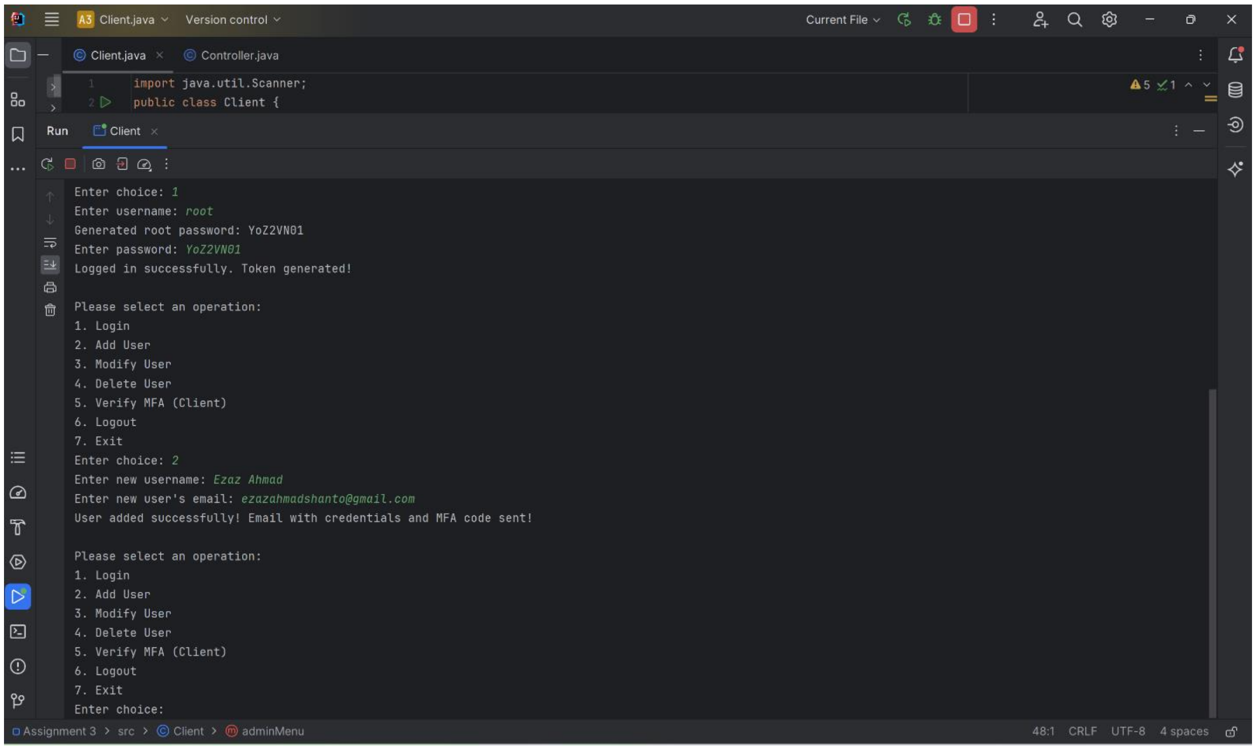

```
import java.util.Scanner;
public class Client {
```

```
1. Admin
2. Client
3. Exit
Enter choice: 1
Welcome to the Admin Console.

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice: 1
Enter username: root
Generated root password: YoZ2VN01
Enter password: YoZ2VN01
Logged in successfully. Token generated!

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
```

- First admin will log when admin write “root” as a user name then system will share the password by which user can log in

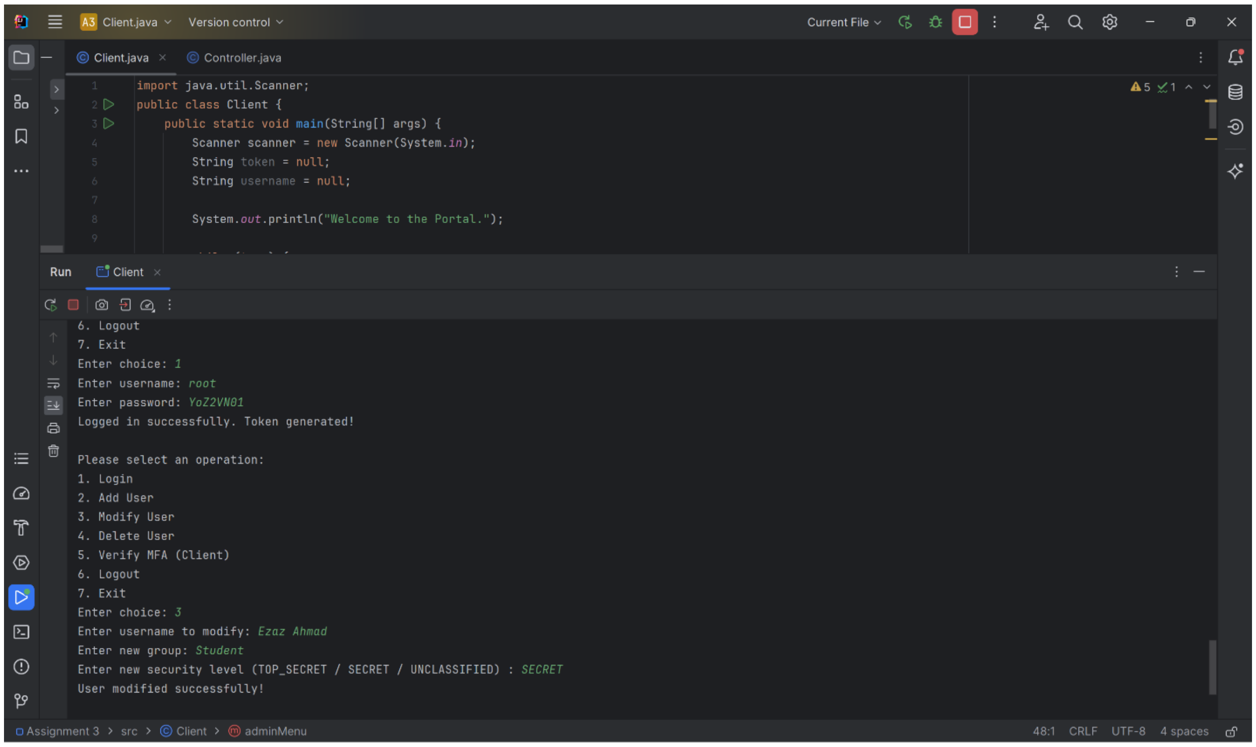

```
import java.util.Scanner;
public class Client {
```

```
Enter choice: 1
Enter username: root
Generated root password: YoZ2VN01
Enter password: YoZ2VN01
Logged in successfully. Token generated!

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice: 2
Enter new username: Ezaz Ahmad
Enter new user's email: ezazahmadshanto@gmail.com
User added successfully! Email with credentials and MFA code sent!

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice:
```

- After that “root” user will add user to the system by typing their name and their email after that system will send the user name and password with the MFA code to that user email address by which user can log into the system.



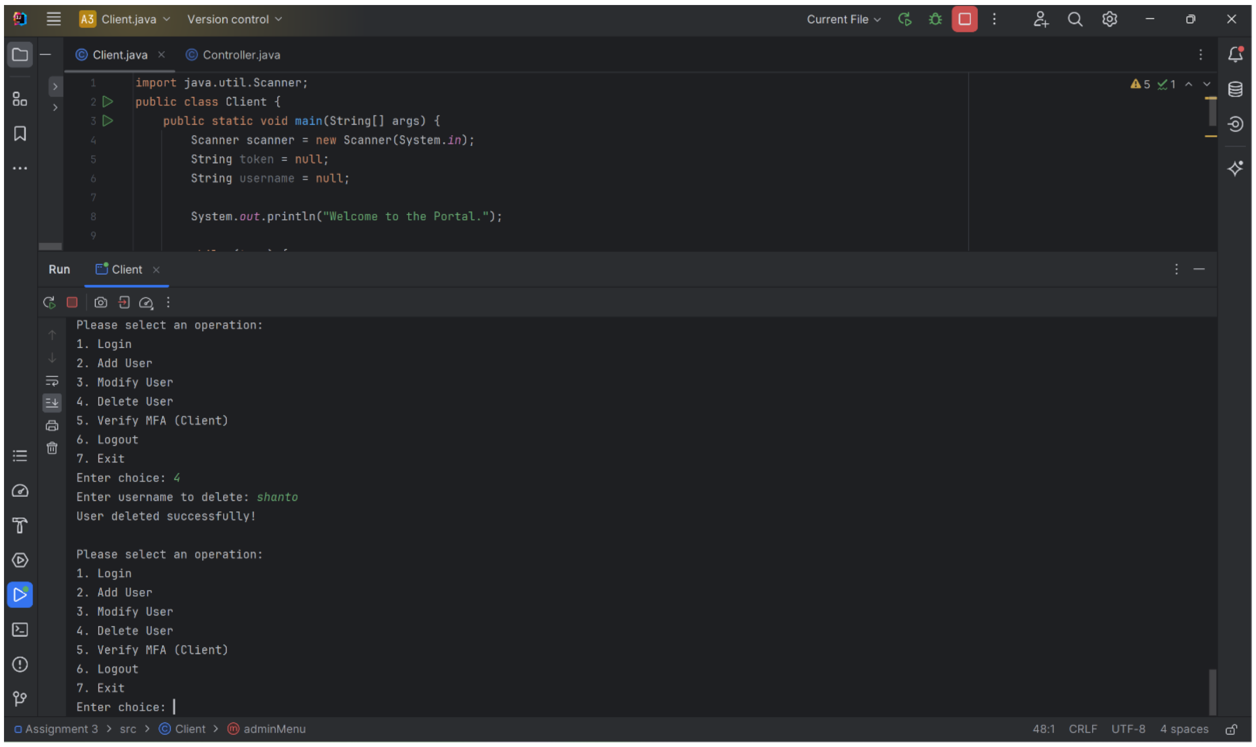
```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
6. Logout
7. Exit
Enter choice: 1
Enter username: root
Enter password: YoZ2VN01
Logged in successfully. Token generated!

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice: 3
Enter username to modify: Ezaz Ahmad
Enter new group: Student
Enter new security level (TOP_SECRET / SECRET / UNCLASSIFIED) : SECRET
User modified successfully!
```

- Here the “root” user is modifying a specific user and assigning new security level.



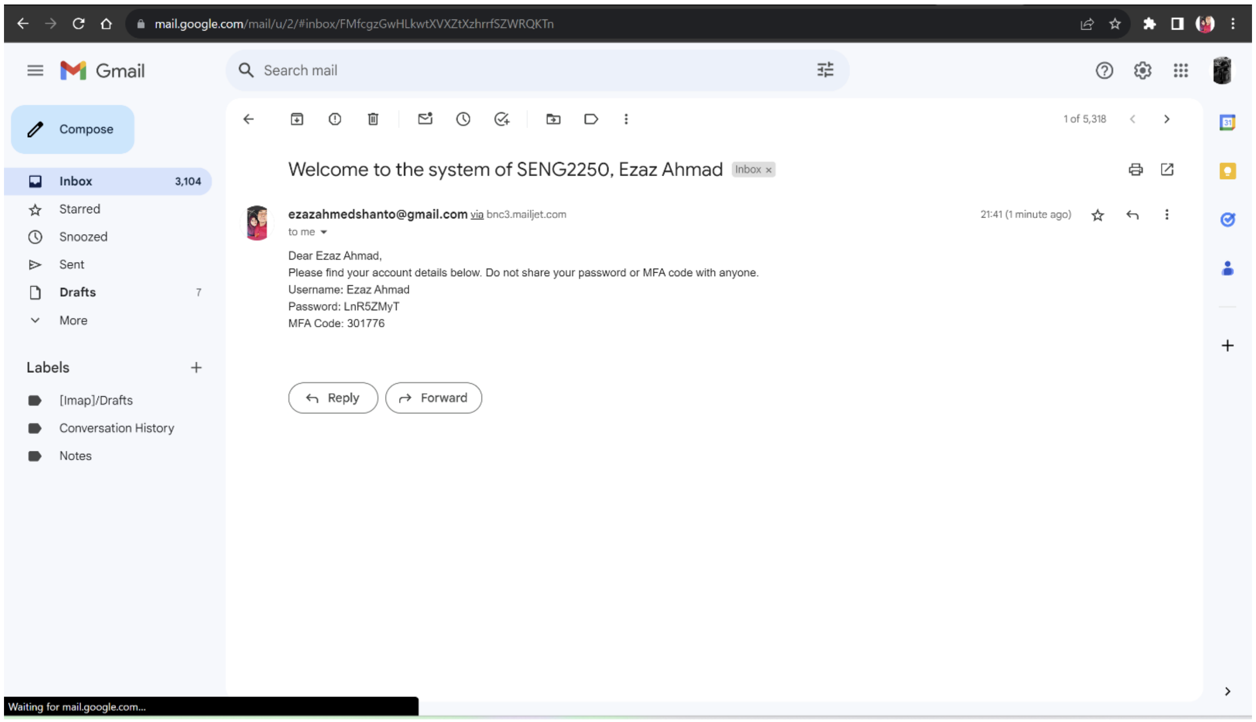
```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice: 4
Enter username to delete: shanto
User deleted successfully!

Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice:
```

- “root” user have the capability to delete a specific user by entering their username


Welcome to the system of SENG2250, Ezaz Ahmad

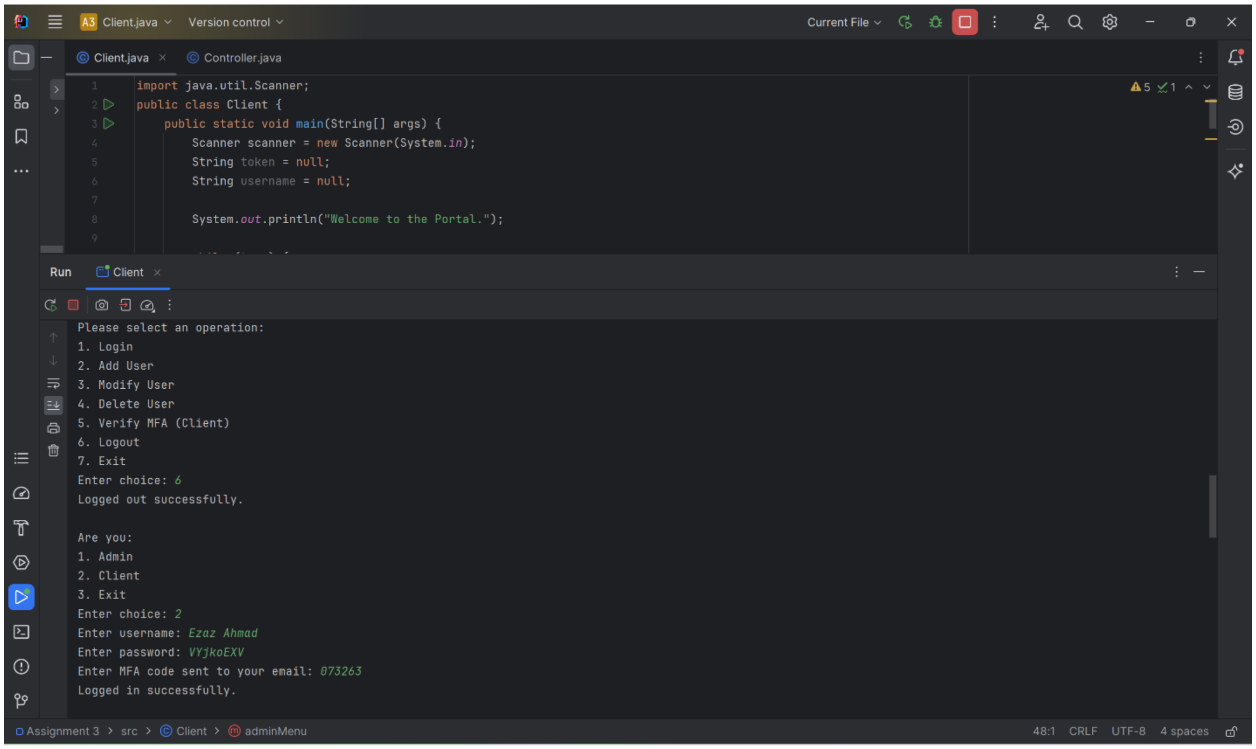
ezazahmedshanto@gmail.com via bnc3.mailjet.com

to me

21:41 (1 minute ago)

Dear Ezaz Ahmad,
Please find your account details below. Do not share your password or MFA code with anyone.
Username: Ezaz Ahmad
Password: LnR5ZMyT
MFA Code: 301776

- This is the email that the user got from the System, where he can get his username, password, MFA code.



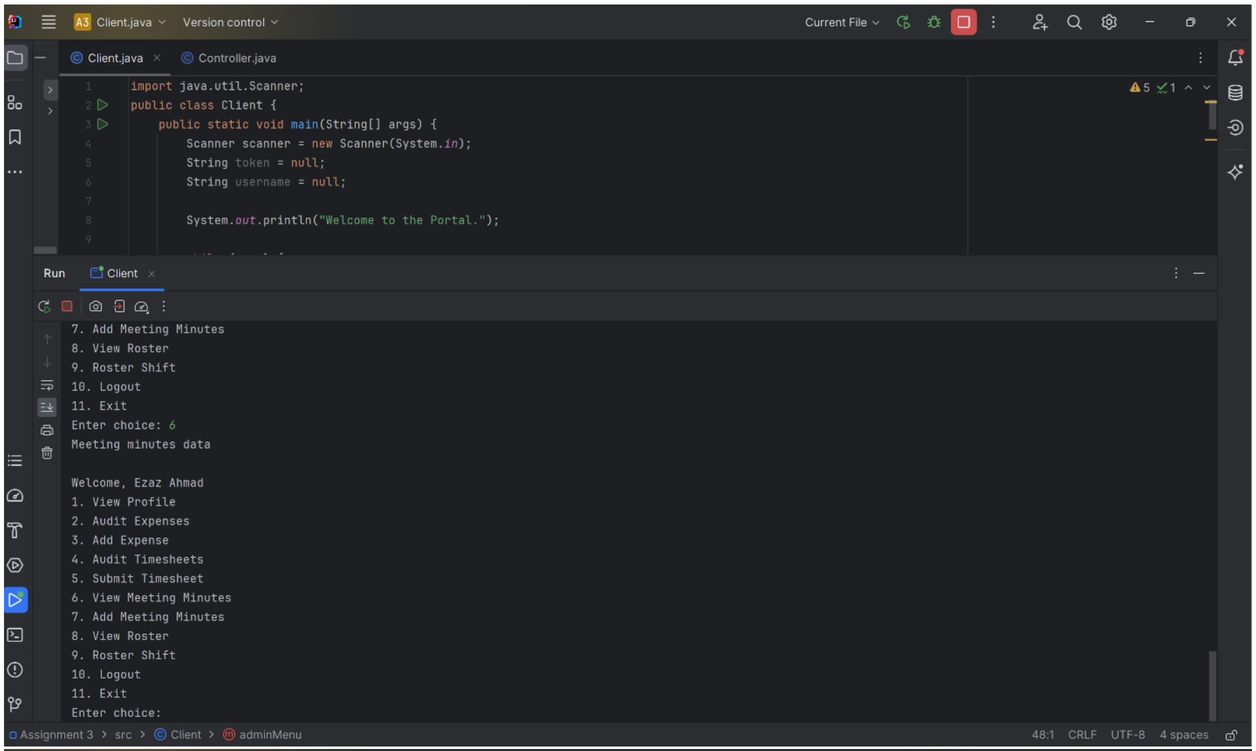
```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
Please select an operation:
1. Login
2. Add User
3. Modify User
4. Delete User
5. Verify MFA (Client)
6. Logout
7. Exit
Enter choice: 6
Logged out successfully.

Are you:
1. Admin
2. Client
3. Exit
Enter choice: 2
Enter username: Ezaz Ahmad
Enter password: VYjkoEXV
Enter MFA code sent to your email: 073263
Logged in successfully.
```

- Using those crediantials that the user got in his email user is logging in his portal

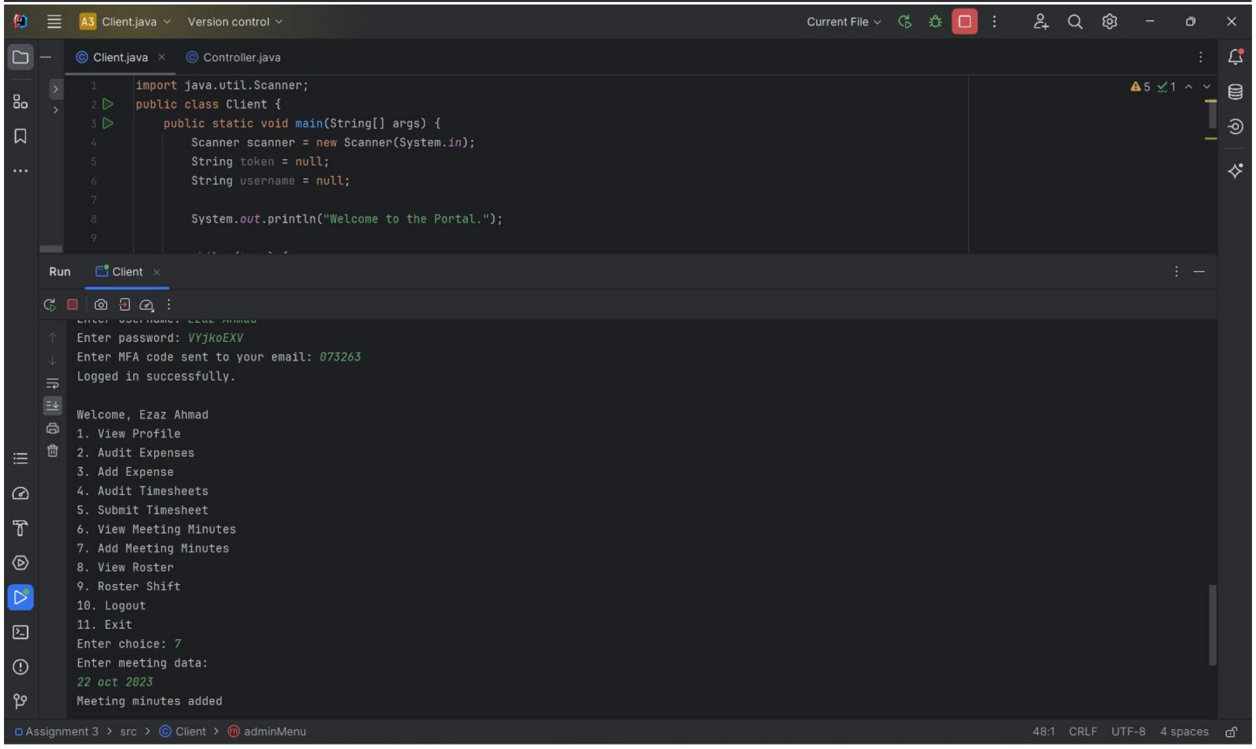


```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
7. Add Meeting Minutes
8. View Roster
9. Roster Shift
10. Logout
11. Exit
Enter choice: 6
Meeting minutes data

Welcome, Ezaz Ahmad
1. View Profile
2. Audit Expenses
3. Add Expense
4. Audit Timesheets
5. Submit Timesheet
6. View Meeting Minutes
7. Add Meeting Minutes
8. View Roster
9. Roster Shift
10. Logout
11. Exit
Enter choice:
```



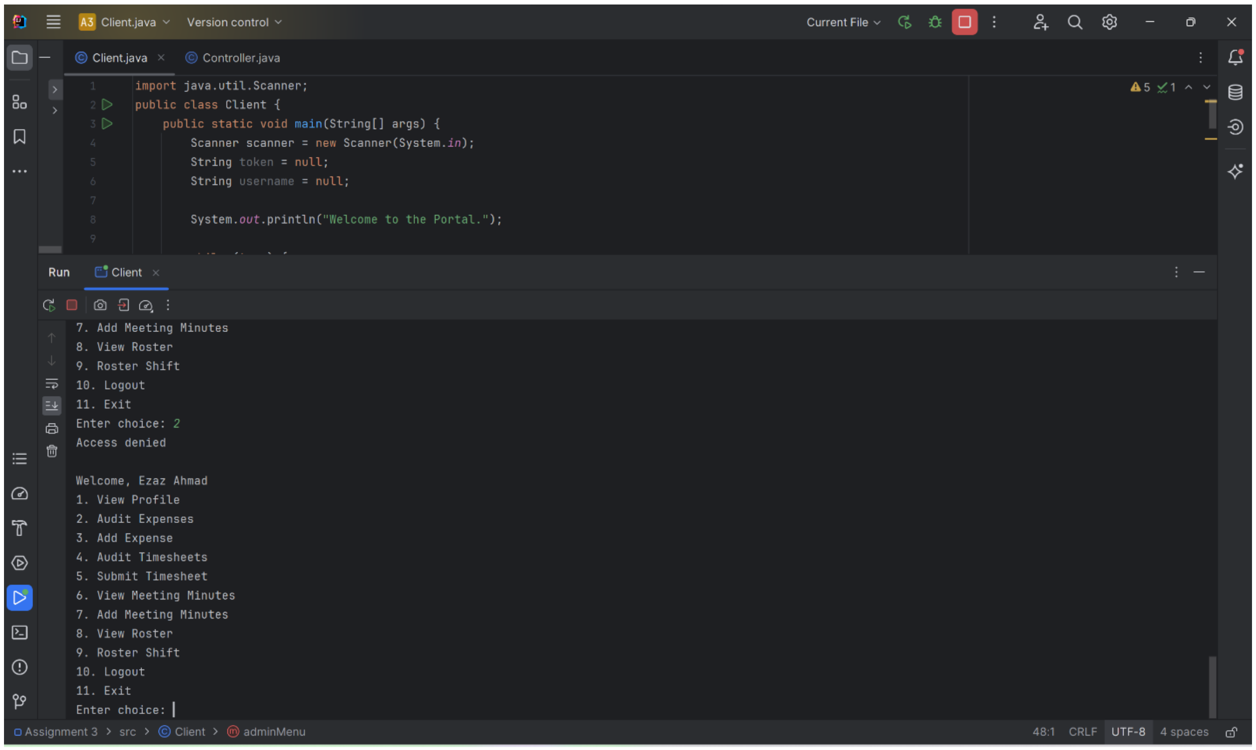
```java
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
Enter password: VYjkoEXV
Enter MFA code sent to your email: 073263
Logged in successfully.

Welcome, Ezaz Ahmad
1. View Profile
2. Audit Expenses
3. Add Expense
4. Audit Timesheets
5. Submit Timesheet
6. View Meeting Minutes
7. Add Meeting Minutes
8. View Roster
9. Roster Shift
10. Logout
11. Exit
Enter choice: 7
Enter meeting data:
22 oct 2023
Meeting minutes added
```

- As this the root user assigned this user as "SECRET" so that's why he is allowed to "view_meeting_minutes" and "add_meeting_minutes". Without this if this user try to access anything else System will stop him and say "Access Denied"



```
import java.util.Scanner;
public class Client {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String token = null;
        String username = null;

        System.out.println("Welcome to the Portal.");
```

```
7. Add Meeting Minutes
8. View Roster
9. Roster Shift
10. Logout
11. Exit
Enter choice: 2
Access denied

Welcome, Ezaz Ahmad
1. View Profile
2. Audit Expenses
3. Add Expense
4. Audit Timesheets
5. Submit Timesheet
6. View Meeting Minutes
7. Add Meeting Minutes
8. View Roster
9. Roster Shift
10. Logout
11. Exit
Enter choice:
```

- System is showing access denied as this user was trying to access to “Audit Expenses” which is not belongs to this user.