UGANDA CHRISTIAN UNIVERSITY

A Center of Excellence in the Heart of Africa

UGANDA CHRISTIAN UNIVERSITY
FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY
DEPARTMENT OF COMPUTING AND TECHNOLOGY
Code Review and Refactoring Challenge

February 14, 2024

NAME	ACCESS NO	REG NO.	COURSE
CHARLES MUGANGA	A96447	J22B23/032	BSCS
TRACY NAJJOBA	A95681	S21B23/034	BSCS
MOGA MUZAMIL ABDUL WAHAB	A94166	S21B23/013	BSCS
ISAAC AROU MAYOL	A94165	S21B23/012	BSCS

Code Review

The following are the Identified Issues from the given classes;

1. **Hardcoded Credentials:**
   - The **login** method uses hardcoded credentials (”admin”, ”admin”), which is a security risk and poor practice.
2. **Password Strength Validation:**
   - The **register** method only checks for the password length, not its complexity or strength, which could be a security concern.
3. **Lack of Input Validation:**
   - Neither **login** nor **register** methods perform any form of input validation (e.g., sanitizing inputs to prevent SQL injection or ensuring the username/email format is valid).
4. **Error Handling:**
   - The methods return plain strings for both success and error messages, which could limit error handling flexibility in a real-world application.
5. **Readability and Maintainability Issues:**
   - Lack of comments explaining the logic, especially in more complex scenarios that might be added later.
   - Magic strings (”Login successful”, ”Invalid credentials”, etc.) are used directly in the code, making it harder to maintain and localize for different languages.

Prioritization

The issues above are prioritized as follows based on their impact on code quality and security:

1. Hardcoded Credentials
2. Lack of Input Validation
3. Password Strength Validation
4. Error Handling
5. Readability and Maintainability Issues

Refactoring

Proposed Changes

Basing on the observed issues in our code review, we have decided to refactor the following implementations to improve its readability, maintainability, and security such that it can be easily understood and maintained by other developers.

1. **Removing Hardcoded Credentials:**
   - Implementing a mechanism to verify credentials against a database or a secure storage solution.
2. **Improving Password Validation:**
   - Enhancing the password validation logic to check for complexity (including uppercase, lowercase, numbers, and special characters).
3. **Input Validation:**
   - Sanitizing and validating inputs to ensure they meet expected formats and criteria, reducing the risk of injection attacks and logical errors.
4. **Error Handling:**
   - Using exceptions or a structured response format (forexample, an array with status and message keys) for error handling and feedback.
5. **Improving Readability and Maintainability:**
   - Adding comments, explaining the purpose and logic of each method.
   - Defining constants for repeated strings to facilitate maintenance and localization.

Link to github repository having all the changes