Daily Coding Assignment Wipro NGA .NET Cohort

Coding Assignment: Secure and Reliable .NET Application

Duration: 90 minutes

Objective: Implement a simple .NET application that demonstrates best practices in security, reliability, and error handling. Write unit tests to validate the implementation.

Problem Statement

You are tasked with developing a basic .NET application that includes secure user authentication, data encryption, and reliable error handling. The application should be a simple user management system with the following features:

1. **User Authentication**: Users should be able to register and log in securely.
2. **Data Encryption**: Sensitive user data should be encrypted.
3. **Error Handling and Logging**: Implement proper error handling and logging mechanisms.

User Stories and Expectations

User Story 1: User Authentication

- **As a user**, I want to be able to register and log in securely.
- **Acceptance Criteria**:
    - The system should allow users to register with a username and password.
    - Passwords should be hashed before storing them in the database.
    - The system should allow users to log in using their username and password.
    - On login, the system should verify the hashed password and authenticate the user.

User Story 2: Data Encryption

- **As a developer**, I want to ensure that sensitive user data is encrypted.
- **Acceptance Criteria**:
    - User passwords should be hashed using a secure hashing algorithm (e.g., SHA-256).
    - Implement encryption for any sensitive data stored in the application (e.g., user details).
    - Use .NET libraries for encryption (e.g., `System.Security.Cryptography`).

User Story 3: Error Handling

- **As a developer**, I want to handle errors reliably and log them appropriately.
- **Acceptance Criteria**:
  - Implement try-catch blocks to handle potential exceptions.
  - Log errors to a file or a logging service (e.g., using `NLog`, `Serilog`, or `log4net`).
  - Ensure that sensitive information is not exposed in error messages.

User Story 4: Logging

- **As a developer**, I want to monitor the application health and track errors through logging.
- **Acceptance Criteria**:
  - Implement logging for successful operations and errors.
  - Log relevant information such as timestamps, error messages, and stack traces.
  - Ensure that logs are saved to a file or a logging service for later review.

Coding Assignment

Task 1: Implement User Authentication

1. **Create a User Model**:
   - Properties: `Username`, `HashedPassword`
   - Methods: `Register()`, `Authenticate()`
2. **Password Hashing**:
   - Use a secure hashing algorithm (e.g., SHA-256) to hash passwords before storing them.
3. **Login System**:
   - Create a login method that verifies the hashed password and authenticates the user.

Task 2: Implement Data Encryption

1. **Encrypt Sensitive Data**:
   - Use .NET's encryption libraries to encrypt sensitive user details (e.g., AES encryption).
2. **Decrypt Data**:
   - Implement a method to decrypt data when needed.

Task 3: Implement Error Handling and Logging

1. **Error Handling**:
   - Use try-catch blocks to handle exceptions and ensure that errors are managed gracefully.
2. **Logging**:
   - Set up a logging framework (e.g., `NLog`, `Serilog`, or `log4net`).
   - Log errors and important events to a file or logging service.

Task 4: Write Unit Tests

1. **Test User Authentication**:
   - Verify that users can register and log in correctly.
   - Ensure that passwords are hashed and verified properly.
2. **Test Data Encryption**:
   - Verify that sensitive data is encrypted and decrypted correctly.
3. **Test Error Handling**:
   - Simulate errors and ensure that they are handled and logged appropriately.
4. **Test Logging**:
   - Verify that logs are written correctly and contain relevant information.

Expectations

- **Code Quality**: Your code should follow best practices for security, reliability, and error handling.
- **Unit Tests**: Ensure that all tests are written to cover the scenarios described and are passing.
- **Documentation**: Include comments and documentation where necessary to explain your code and tests.
- **Submission**: Submit your implementation of the application and unit tests as a single project.

This assignment will assess your ability to design a secure and reliable .NET application, implement secure coding practices, and handle errors effectively.