### User Authentication

#### Test Cases

##### 1. Login with Valid Credentials

* **Description**: Verify that the user can log in with valid credentials.
* **Steps**:
  1. Send a POST request to /api/Auth/login with a JSON containing the valid username and password.
  2. Verify that the response has a status code of 200 OK.
  3. Ensure that a valid authentication token is received.
* **Test Data**:

json

{

"username": "testuser",

"password": "password"

}

* **Expected Result**: Response with status code 200 and a valid authentication token.

##### 2. Error Handling for Invalid Credentials

* **Description**: Verify that login attempts with invalid credentials are handled correctly.
* **Steps**:
  1. Send a POST request to /api/Auth/login with invalid credentials.
  2. Verify that the response has a status code of 401 Unauthorized.
  3. Ensure that the error message is appropriate for invalid credentials.
* **Test Data**:

json

{

"username": "invaliduser",

"password": "wrongpassword"

}

* **Expected Result**: Response with status code 401 and an appropriate error message.

##### 3. Token Generation and Expiration

* **Description**: Verify that the authentication token is generated correctly and expires according to the defined time.
* **Steps**:
  1. Log in with valid credentials and obtain a token.
  2. Verify the token's validity during its lifetime.
  3. Wait for the token to expire and verify that the token is no longer valid.
* **Test Data**:

json

{

"username": "testuser",

"password": "password"

}

* **Expected Result**: Token is valid during its lifetime and expires correctly after the defined time.