Introduction

This document outlines the suite of API test cases for the core authentication module (auth.py) of the Hammer & Grammar Management System. The purpose of these tests is to verify the correctness, security, and robustness of the endpoints responsible for user authentication.

The test cases cover the three primary features of the authentication system:

1. **User Login**: Validating user credentials and issuing JSON Web Tokens (JWTs).
2. **Token Refresh**: Ensuring that valid refresh tokens can be used to obtain new access tokens.
3. **User Logout**: Confirming that tokens are successfully invalidated upon logout by adding them to a blocklist.

Each test is designed to be executed in a Postman environment and includes positive and negative scenarios to ensure comprehensive coverage.

Testing Environment

- **Tool**: Postman
- **Base URL**: http://127.0.0.1:5000 (or the relevant development server address)
- **Prerequisites**: The application is running, and the database has been seeded with the sample data provided in *seed.sql*. This ensures that user accounts exist for testing.

Feature: User Login

- **Feature ID**: AUTH-01
- **Endpoint**: POST /auth/login
- **Description**: Verifies the functionality of the user login endpoint, which validates credentials and returns JWT access and refresh tokens.
- **Test Cases**:

Test Case ID	Description	Test Steps	Expected Result (ER)

AUTH-01-TC-001	**Successful Login**
	(User provides valid credentials)
		1. Set method to POST.
		2. Set URL to http://127.0.0.1/5000/auth/login
		3. Set Body to raw JSON:
		{
		“username” : “NMKhoi”,
		“password”:”123456789”
		}
		4. Send request.
			ER: Status code is 200 OK. The response body is a JSON object containing non-empty access_token and refresh_token strings.

AUTH-01-TC-002	**Invalid Password**
	(User provides a correct username but an incorrect password)
		1. Set method to POST.
		2. Set URL to http://127.0.0.1/5000/auth/login
		3. Set *content-type* header to *application/json*.
		4. Set Body to raw JSON:
		{
		“username” : “NMKhoi”,
		“password”:”12345”
		}
		5. Send request.
			ER: Status code is **401 Unauthorized**. The response body contains the message: Invalid credentials.

AUTH-01-TC-003	**Invalid Username** (User provides a non-existent username)	1. Set method to POST.
		2. Set URL to http://127.0.0.1/5000/auth/login
		3. Set *content-type* header to *application/json*.
		4. Set Body to raw JSON:
		{
		"username" : "NMK",
		"password":"123456789"
		}
		5. Send request.
			ER: Status code is **401 Unauthorized**. The response body contains the message: Invalid credentials.

AUTH-01-TC-004	**Missing Credentials** (User provides an empty JSON object)	1. Set method to POST. 2. Set URL to http://127.0.0.1/5000/auth/login 3. Set *content-type* header to *application/json*. 4. Set Body to raw JSON: {} 5. Send request.	**ER**: Status code is **400 Bad Request**. The response body contains a validation error message indicating that *username* and *password* are required fields
AUTH-01-TC-005	**Empty Credentials** (User provides empty strings for credentials)	1. Set method to POST. 2. Set URL to http://127.0.0.1/5000/auth/login	**ER**: Status code is **400 Bad Request**. The response body contains the

		3. Set *content-type* header to *application/json*.
		4. Set Body to raw JSON:
		{
		"username": "",
		"password": ""
		}
		5. Send request.
			message: *Username* and *password* are required.

AUTH-01-TC-006	**Non-JSON Request** (Request is sent without the application/json content type)	1. Set method to POST. 2. Set URL to http://127.0.0.1/5000/auth/login 3. Do not set the *content-type* header. 4. Send request.	**ER:** Status code is **400 Bad Request**. The response body contains an error message related to invalid input or JSON format.

- **Results:**

Test Case ID	Result	Status	Notes
AUTH-01-TC-001	Similar to ER	PASSED	N/A
AUTH-01-TC-002	Similar to ER	PASSED	N/A
AUTH-01-TC-003	Similar to ER	PASSED	N/A

AUTH-01-TC-004	Similar to ER	PASSED	N/A
AUTH-01-TC-005	Similar to ER	PASSED	N/A
AUTH-01-TC-006	• **500 Internal Server Error** • "error": "415 Unsupported Media Type: Did not attempt to load JSON data because the request Content-Type was not 'application/json'.", • "message": "Unexpected error occured"	FAILED	See BUG-001

Feature: Access Token Refresh

- **Feature ID:** AUTH-02
- **Endpoint:** POST /auth/refresh
- **Description:** Verifies that a client can use a valid refresh token to obtain a new access token.
- **Test Cases:**

Test Case ID	Description	Test Steps	Expected Result (ER)
AUTH-02-TC-001	**Successful Refresh** (User provides a valid refresh token)	1. Perform a successful login (AUTH-01-TC001) and store the *refresh_token*. 2. Set method to POST. 3. Set URL to http://127.0.0.1/5000/auth/refresh 4. Set Authorization header to Bearer {{refreshToken}}. 5. Send request.	**ER**: Status code is **200 OK**. The response body is a JSON object containing a new, non-empty *access_token* string.
AUTH-02-TC-002	**Refresh with Access Token** (User attempts to use an access token to refresh)	1. Perform a successful login (AUTH-01-TC001) and store the *access_token*. 2. Set method to POST. 3. Set URL to http://127.0.0.1/5000/auth/refresh 4. Set Authorization header to Bearer {{accessToken}}. 5. Send request.	**ER**: Status code is **401 Unauthorized**. The response body contains a message indicating that only refresh tokens are allowed.
AUTH-02-TC-003	**Refresh with Invalid Token** (User provides a malformed or expired token)	1. Set method to POST. 2. Set URL to http://127.0.0.1/5000/auth/refresh 3. Set *Authorization* header to Bearer an-invalid-token. 4. Send request.	**ER**: Status code is **401 Unauthorized**. The response body contains a message like *Token is invalid or expired*.

AUTH-02-TC-004	**Refresh without Token** (User makes a request with no Authorization header)	1. Set method to POST. 2. Set URL to http://127.0.0.1/5000/auth/refresh 3. Do not include an *Authorization* header. 4. Send request.	**ER**: Status code is **401 Unauthorized**. The response body contains a message like *Missing Authorization Header*.

- **Results:**

Test Case ID	Result	Status	Notes
ATUH-02-TC-001	Similar to ER	PASSED	N/A
AUTH-02-TC-002	• **422 Unprocessable Entity** • **"msg":** "Only refresh tokens are allowed"	FAILED	See BUG-002
AUTH-02-TC-003	• **422 Unprocessable Entity** • **"msg"**: "Signature verification failed" or "Invalid crypto padding."	FAILED	See BUG-003
AUTH-02-TC-004	Similar to ER	PASSED	N/A

Feature: User Logout

- **Feature ID:** AUTH-03
- **Endpoint:** DELETE /auth/logout
- **Description:** Verifies that a user can securely log out, which invalidates their current JWTs by adding them to a blocklist

Test Case ID	Description	Test Steps	Expected Result (ER)
AUTH-03-TC-001	**Successful Logout** (User provides a valid access token to log out)	1. Perform a successful login (AUTH-01-TC001) and store the access_token. 2. Set method to DELETE. 3. Set URL to http://127.0.0.1/5000/auth/logout. 4. Set Authorization header to Bearer {{accessToken}}. 5. Send request.	**ER**: Status code is **200 OK**. The response body contains the message: Successfully logged out!.
AUTH-03-TC-002	**Verify Token is Blocklisted** (Attempt to use a token immediately after logout)	1. Complete all steps for AUTH-03-TC001. 2. Using the **same** access_token from step 1, immediately attempt to call the /auth/logout endpoint again. 3. Send request.	**ER**: Status code is **401 Unauthorized**. The response body contains the message: *Token has been revoked*.
AUTH-03-TC-003	**Logout with Invalid Token** (User provides a malformed or expired token)	1. Set method to DELETE. 2. Set URL to http://127.0.0.1/5000/auth/logout. 3. Set Authorization header to Bearer an-invalid-token.	**ER**: Status code is **401 Unauthorized**. The response body contains a message like

		4. Send request.	*Token is invalid or expired*.
AUTH-03-TC-004	**Logout without Token** (User makes a request with no Authorization header)	1. Set method to DELETE. 2. Set URL to http://127.0.0.1/5000/auth/logout. 3. Do not include an *Authorization* header. 4. Send request.	**ER**: Status code is **401 Unauthorized**. The response body contains a message like *Missing Authorization Header*.

- **Results:**

Test Case ID	Result	Status	Notes
AUTH-03-TC-001	Similar to ER	PASSED	N/A
AUTH-03-TC-002	Similar to ER	PASSED	N/A
AUTH-03-TC-003	• **422 Unprocessable Entity** • **"msg"**: "Signature verification failed" or “Invalid crypto padding.”	FAILED	See BUG-004

AUTH-03-TC-004	Similar to ER	PASSED	N/A