Master

Test Cases

API 0

Users 0

Signup 7

C73: Successful Registration with Valid Credentials

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Steps

1. Send a POST request to the /api/users/signup endpoint with a valid email and password in the request body. Ensure the email is not already in use.
2. Check the response body.

Expected Result

The API should return a 201 Created status code. The response body should contain a JSON object with user information:

{

   "user": {

      "username": «username»,

      "email": «username@gmail.com»,

      "dailyWaterNorm": 2000,

      "weight": 0,

      "sportTime": 0

   }

}.

The response body should contain the email address provided during registration, and the username should contain a truncated email (before the '@' symbol, excluding the symbol itself).

C75: Registration Failure - Missing Email Field

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Steps

1. Send a POST request to the /api/users/signup endpoint with a request body that is missing the 'email' field but includes the 'password' field.

Expected Result

The API should return a 400 Bad Request error. The response body should contain an error message.

C76: Registration Failure - Missing Password Field

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Steps

1. Send a POST request to the /api/users/signup endpoint with a request body that includes only the 'email' field but is missing the 'password' field.

Expected Result

The API should return a 400 Bad Request. The response body should contain an error message.

C77: Registration Failure - Invalid Email Format

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Steps

1. Send a POST request to the /api/users/signup endpoint containing an invalid email format and a valid password.

The API should return a 400 Bad Request error. The response body should contain an error message.

C78: Registration Failure - Email Already Exists

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify the user.

Steps

1. Send a POST request to the /api/users/signup endpoint with a valid email and password that already exists in the system.

Expected Result

The API should return a 409 Conflict. The response body should contain an error message.

C79: Registration with too Short Password

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Steps

1. Send a POST request to /api/users/signup with a valid email and a short password (less than 8 characters).

Expected Result

The response status code should be 400. The response body should contain an error message.

C80: Registration Failure - Additional Fields

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			

Test Case Labels

Authentication

Steps

1. Send a POST request to /api/users/signup with a valid email, a strong password, and additional fields (e.g., username).

Expected Result

The API should return a 400 Bad Request. The response body should contain an error message.

Signin   7

C58: Successful Sign In with Valid Credentials

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint with valid email and password for a verified user.

Expected Result

The server responds 200 OK.

The response body contains a JSON object with a valid JWT token and user data:

{

    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjY5N2UxYzM2ZGQ2ZjMxMzdjMzYxYzk0ZSIsImlhdCI6MTc2OTg3NDM4OCwiZXhwIjoxNzY5OTU3MTg4fQ.FYayorBX6Rx1Ik34ZQmmELd8uQQH0bO2D87kUfY5EhI",

    "user": {

        "username": "username",

        "email": "username@gmail.com",

        "dailyWaterNorm": 2000,

        "weight": 0,

```
        "sportTime": 0

    }

}
```

C63: Sign In with Invalid Email

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint.
2. In the request body, provide a valid password and an invalid email address.

Expected Result

The response status code is 401 Unauthorized. The response body should contain an error message.

C61: Sign In with Incorrect Password

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint.
2. In the request body, provide an invalid password and a valid email address.

Expected Result

The response status code is 400 Bad Request. The response body should contain an error message.

C62: Sign In with Unverified Email

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up but do not verify the user.

Steps

1. Send a POST request to the /api/users/signin endpoint.
2. In the request body, provide a valid password and a valid email address.

Expected Result

The response status code is 401 Unauthorized.

The response body should contain an error message.

User should NOT be granted access to the system.

User should NOT receive an authentication token.

C59: Sign In with Missing Email

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint with a request body that is missing the 'email' field, but includes the 'password' field.

Expected Result

The API should return a 400 Bad Request. The response body should contain an error message.

C60: Sign In with Missing Password

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint with a body that includes only the 'email' field, omitting the 'password' field. Use a valid email address for an existing user.

Expected Result

The API should return a 400 Bad Request. The response body should contain an error message.

C64: Sign In with Empty Email

Type	Priority	References	Automation Type
Functional	High	None	None
Is Automated			
No			
Test Case Labels			
Authentication			

Preconditions

Sign up and verify user.

Steps

1. Send a POST request to the /api/users/signin endpoint with an empty string for the email field and a valid password in the request body.

Expected Result

The response status code should be 400 Bad Request. The response body should contain an error message.

Signout 2

C71: Successful Sign Out

Type	Priority	References	Automation Type
Functional	Medium	None	None
Is Automated			
No			

Test Case Labels

Authentication

Preconditions

Sign up and verify the user.

Sign in with valid credentials.

Steps

1. Send a POST request to the /api/users/signout endpoint with a valid Bearer token.

Expected Result

The server responds with a 200 OK status code.
The response body contains a message.

C72: Sign Out with Invalid Token

Type	Priority	References	Automation Type
Functional	Medium	None	None

Is Automated

No

Test Case Labels

Authentication

Preconditions

Sign up and verify the user.

Sign in with valid credentials.

Steps

1. Send a POST request to the /api/users/signout endpoint with the invalid JWT token.

Expected Result

The API returns a 401 Unauthorized status code. The response body should contain an error message.