

# NightPhoto App

## User authentication Testing Report

10.8.24 – 18.8.24

Hannu Salmi

---

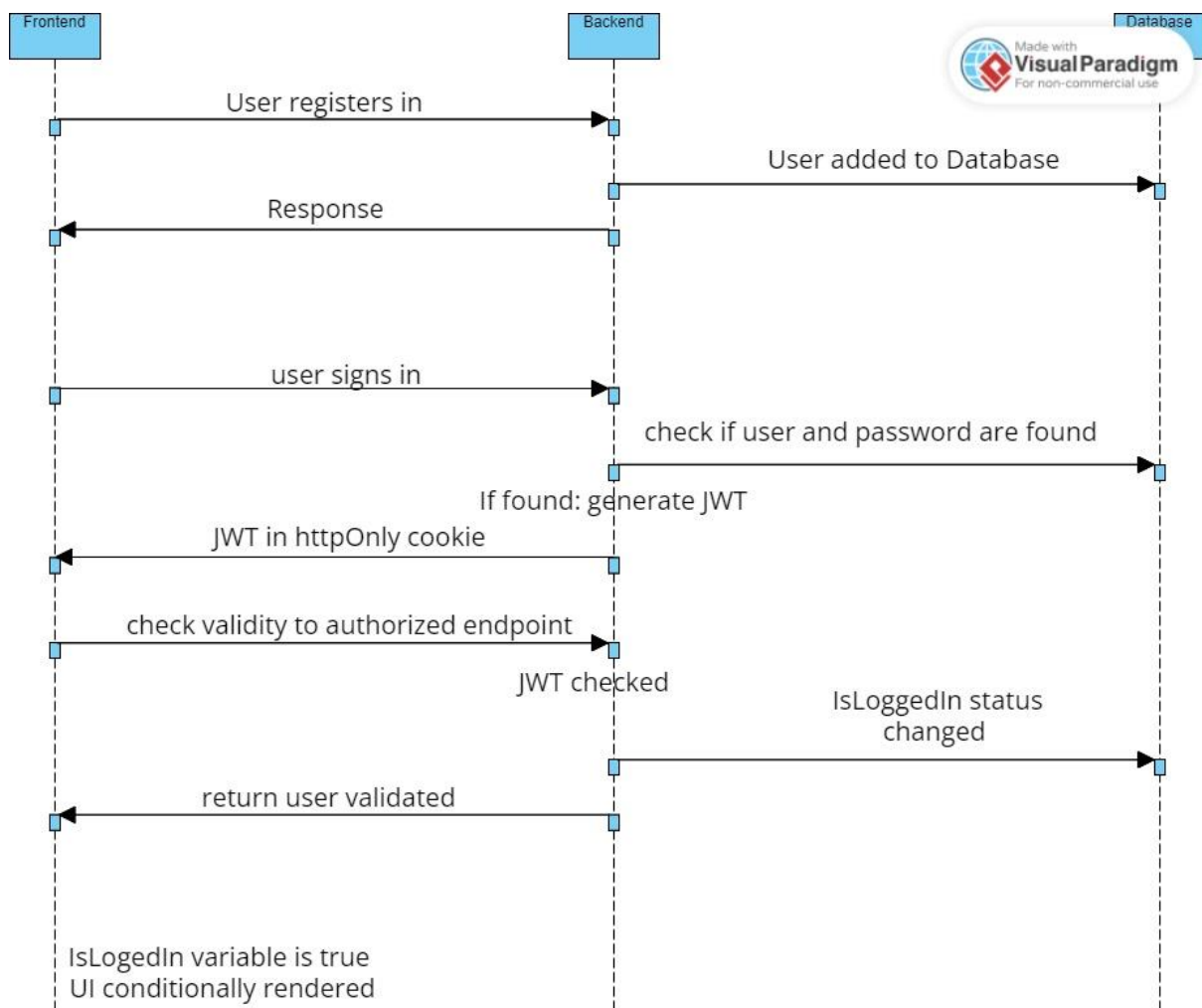
### Table of Contents:

1. [Introduction](#)
  2. [Test Plan Summary](#)
  3. [Test Cases](#)
  4. [Test Results Summary](#)
  5. [Epilogue](#)
-

# 1. Introduction:

**Overview:** NightPhoto App is a full-stack website where users can upload images and comment on them. The development of the project is still ongoing, but some features are already available, and the site is deployed on Azure App Service. One of the available features is user registration and authentication. This document reports the results of the tests conducted on this feature.

**The routine:** The user registration and authentication process operates as follows: The user fills out a registration form, which is then passed to the backend and stored in the database. Later, when the user signs in, the backend checks if the user is found in the database. If the user is found, a JWT is created for the user and returned in an HttpOnly cookie. In the same function on the frontend, when the client receives the JWT response, it makes another call to the backend to an authorized endpoint where the JWT is verified.



## **2. Test Plan Summary:**

**Objectives:** The test is divided into two parts: the first part focuses on the user registration process, and the second part addresses the user sign-in method.

**Scope:** User registration and user sign in.

### **Registration Testing:**

Using Robot Framework for end-to-end testing, we create and test users in various ways:

- Valid Registration

- Empty Fields

- Invalid Email Format

- Duplicate Username

### **Authentication:**

- Valid Credentials

- Invalid Credentials

- Authorized Endpoint Access

**Environment and Tools:** Robot framework

3. Test Cases:

Project Name: NightPhoto App

User Validation Testing

Test Case ID: reg\_1

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Register Page

Test Title: User registration with valid details

Description: Registering User with valid details

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app /Register	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register	The Register page should exist and be found	Page found	Pass	
2	Try to create user with valid details	firstname:testdata lastname:test username:data email:testdata@example.com password:testing123	The registration should return 201 Created	Post request returns 201 Created	Pass	
3	Verify that user data exists in DB		The data should be found in NightPhotoDB	Data found	Pass	

Post-conditions:  
The user is created using valid details.

Project Name: NightPhoto App

User Validation Testing

Test Case ID: reg\_2

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Register Page

Test Title: User registration with empty fields

Description: Registering User with empty fields

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app /Register	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register	The Register page should exist and be found	Page found	Pass	
2	Try to create user with empty fields	firstname: lastname: username: email: password:	The registration with empty fields should be denied and error message should appear.	Post request returns 201 Created	Fail	

Post-conditions:  
It is possible to create new user with empty fields.

Project Name: NightPhoto App

## User Registration Testing

Test Case ID: reg\_3

Test Designed by: Hannu Salmi

Test Priority (Low/Medium/High): Medium

Test Designed date: 2.8.2024

Module Name: NightPhoto App Register page

Test Executed by: Hannu Salmi

Test Title: User registration with invalid email format

Test Execution date: 14.8.2024

Description: Registering with invalid email format

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app register user page	<a href="https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register">https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register</a>	User should find the /Register page	/Register page found	Pass	
2	Try to create a user with invalid email format	firstname: Test lastname: Data username: testdata email: invalidemail password: swordfish	a message of invalid email should appear	a message of invalid email appeared	Pass	
3	Verify that user with invalid email format is not in database		User data should not be found in database	User data not found in database	Pass	

Post-conditions:

A new user with invalid email format is not registered to database and a message of invalid email appears if invalid email is given to email field.

Project Name: NightPhoto App

## User Registration Testing

Test Case ID: reg\_4

Test Designed by: Hannu Salmi

Test Priority (Low/Medium/High): Medium

Test Designed date: 2.8.2024

Module Name: NightPhoto App Register page

Test Executed by: Hannu Salmi

Test Title: User registration duplicate username

Test Execution date: 14.8.2024

Description: Registering with duplicate username

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app register user page	<a href="https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register">https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Register</a>	User should find the /Register page	/Register page found	Pass	
2	Try to create a user with username that already exists	firstname: Test lastname: Data username: duplicate email: invalidemail password: swordfish	Should return 409 Conflict	Returns 409 Conflict	Pass	
3	Verify that user with duplicate username is not found in database		Duplicate user data should not be found in database	Duplicate user data not found in database	Pass	

Post-conditions:

A new user with duplicate username is not registered to database and the post request of the form returns 409 Conflict.

Project Name: NightPhoto App

## User Validation Testing

Test Case ID: val\_1

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Login page

Test Title: User validation with username and password

Description: Sign in user with username and password

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app register user page	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Login	User should find the /Login page	Page found	Pass	
2	Try to create a user with username that already exists	username: testdata123 password: testing123	Should redirect user to homepage and a yellow button should appear indicating that user is now signed in.	User is redirected to mainpage and yellow button appears	Pass	

Post-conditions:

User is redirected to mainpage and in the top right corner is a yellow button that indicates that user is now signed in.

Project Name: NightPhoto App

## User Validation Testing

Test Case ID: val\_2

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Login page

Test Title: Users cannot sign in with invalid details

Description: Users cannot sign in with invalid details

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app register user page	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Login	User should find the /Login page	Page found	Pass	
2	create user with invalid details	username: invalidusername password: invalidpswrđ	No yellow button indicating that user is signed in appears.	user is not redirected to mainpage and sign in button does not appear	Pass	

Post-conditions:

User is in signin page and sign in button does not seem to do anything.

Project Name: NightPhoto App

## User Validation Testing

Test Case ID: val\_3

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Upload Image page

Test Title: User, after sign in, can access authorized content on page

Description: User can access authorized content on webpage

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app register user page	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/Login	User should find the /Login page	Page found	Pass	
2	create user with valid details	username: testdata123 password: testing123	User redirected to main page. Yellow button appears to top right corner, indicating that user is signed in.	User redirected to mainpage and button appears	Pass	
3	User clicks Upload Image link on the navbar	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/UploadImage	There should be a possibility to select file and a button to Upload it.	Upload button appears. There is a possibility to choose a file.	pass	

Post-conditions:

User can access content that is authorized for signed in users.

Project Name: NightPhoto App

## User Validation Testing

Test Case ID: val\_4

Test Priority (Low/Medium/High): High

Module Name: NightPhoto App Upload Image page

Test Title: Unvalidated users cannot access content on /UploadImage

Description: Not signed in user can access content on /UploadImage

Test Designed by: Hannu Salmi

Test Designed date: 6.8.2024

Test Executed by: Hannu Salmi

Test Execution date: 17.8.2024

Pre-conditions:

Dependencies:

Step	Test Steps	Test Data	Expected Result	Actual Result	Status (Pass/Fail)	Notes
1	Navigate to NightPhoto app /UploadImage	https://nightphoto-dccta3ate6ajbtb8.northeurope-01.azurewebsites.net/UploadImage	There should be a message indicating that user must sign in before upload of an image is possible.	"Please log in to upload an image." – message appears	Pass	

Post-conditions:

User cannot access authorized content before sign in.

#### 4. Test Results Summary:

All tests passed except for the Empty Fields test case in the registration process. This indicates that the system does not handle empty fields as expected during user registration, which needs to be addressed to ensure robust validation and user experience.

**Number of test cases executed:** 6

**Passed:** 5

**Failed:** 1

**Critical Issues:** User was able to register to the webpage with empty fields.

---

---

```
=====
Register User
=====
Creating a new user to db with valid details | PASS |
-----
Creating a new user with empty fields | FAIL |
'[(3, '', '', '', '', '', '', False), (4, '', '', '', '', '', '', False)]' should be empty.
-----
Creating a new user with invalid email format | PASS |
-----
Creating a new user with duplicate username | PASS |
-----
Sign in with valid details | PASS |
-----
Sign in with invalid details | PASS |
-----
Signed in user can access authorized endpoint | PASS |
-----
User cannot access authorized content before sign in | PASS |
-----
Register User | FAIL |
8 tests, 7 passed, 1 failed
=====
```



## Epilogue:

The error was fixed by adding “Required”-annotations to model and checking the validation in controller:

```
[HttpPost]
public async Task<ActionResult<UserModel>> PostUsersTable(UserModel userTable)
{
    if (!ModelState.IsValid)
    {
        return BadRequest(ModelState);
    }
}
```

```
13 references
public partial class UserModel
{
    7 references
    public int Id { get; set; }

    [Required(ErrorMessage = "Firstname is required")]
    2 references
    public string Firstname { get; set; }

    [Required(ErrorMessage = "Lastname is required")]
    2 references
    public string Lastname { get; set; }

    [Required(ErrorMessage = "Username is required")]
    9 references
    public string Username { get; set; }

    [Required(ErrorMessage = "Email is required")]
    [EmailAddress(ErrorMessage = "Invalid email address")]
    1 reference
    public string Email { get; set; }

    [Required(ErrorMessage = "Password is required")]
    2 references
    public string Password { get; set; }

    3 references
    public string Role { get; set; }

    3 references
    public bool IsLoggedIn { get; set; }
}
```