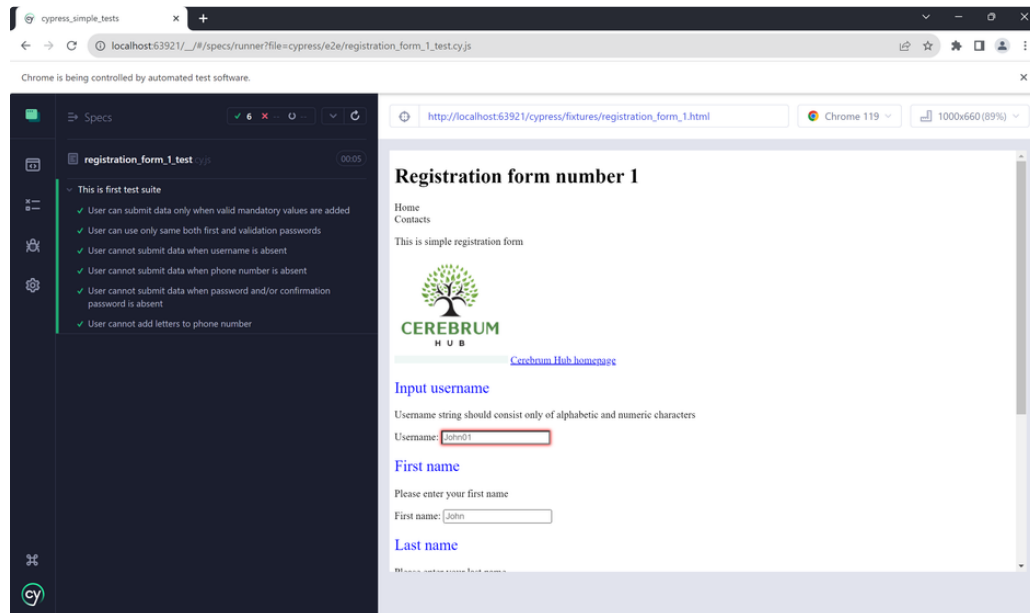


Sigrid Laas-Hurt

- Assignment 1: PREPARE YOUR LOCAL WORKPLACE
- Assignment 2: UPDATE TESTS FOR REGISTRATION FORM 1
- Assignment 3: CREATE NEW TESTS FOR REGISTRATION FORM 1
- BONUS Assignment 6: FIX TEST FILE WITH FAILED TESTS
- BONUS Assignment 7: CREATE VISUAL AND FUNCTIONAL TESTS FOR REGISTRATION FORM 3

Assignment 1: PREPARE YOUR LOCAL WORKPLACE



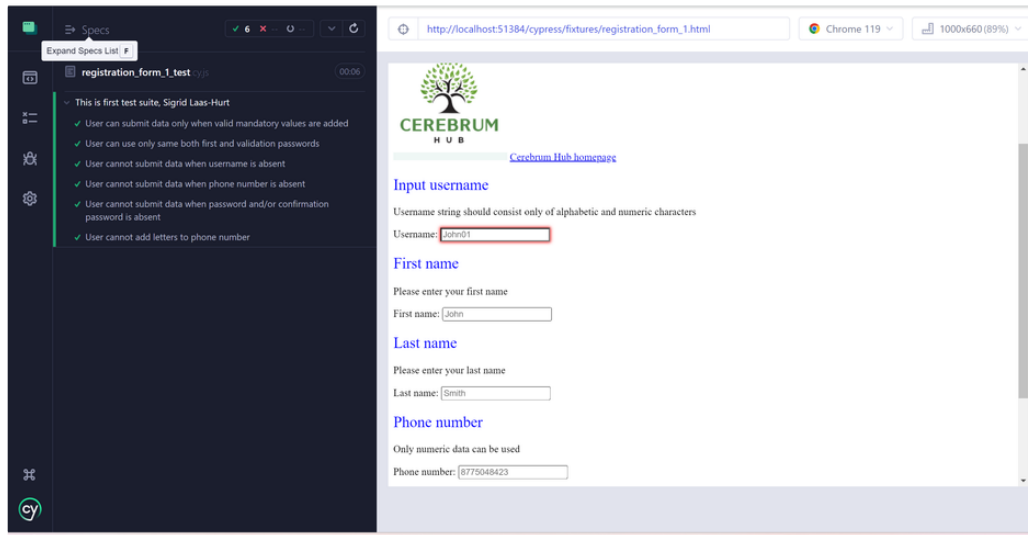
Assignment 2: UPDATE TESTS FOR REGISTRATION FORM 1

```
describe('This is first test suite, Sigrid Laas-Hurt', () => {
  it('User can submit data only when valid mandatory values are added', () => {
    //17.11.2023
    cy.get('[data-testid="phoneNumberTestId"]').type('555666777')
    cy.get('input[name="password"]').type('Password123SLH')
    cy.get('[name="confirm"]').type('Password123SLH')
    cy.get('#username').type('Something')

    //in order to activate submit button, user has to click somewhere outside the input field
    cy.get('h2').contains('Password').click()
    cy.get('.submit_button').should('be.enabled')
    cy.get('.submit_button').click()

    // Assert that both input and password error messages are not shown
    // next 2 lines check exactly the same, but using different approach
    cy.get('#input_error_message').should('not.be.visible')
    cy.get('#password_error_message').should('have.css', 'display', 'none')

    // Assert that success message is visible
    // next 2 lines check exactly the same, but using different approach
    cy.get('#success_message').should('be.visible')
    cy.get('#success_message').should('have.css', 'display', 'block')
  });
});
```



Assignment 3: CREATE NEW TESTS FOR REGISTRATION FORM 1

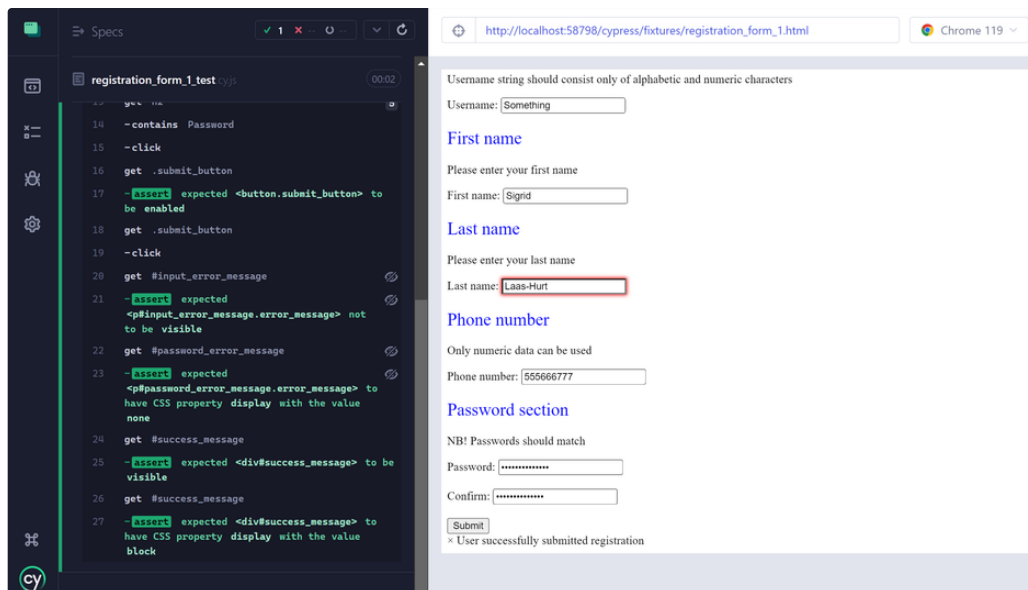
2. Add new steps for filling in the first name and last name to the test 'User can submit data only when valid mandatory values are added'

```
describe('This is first test suite, Sigrid Laas-Hurt', () => {
  it.only('User can submit data only when valid mandatory values are added', () => {
    //17.11.2023
    cy.get('[data-testid="phoneNumberTestId"]').type('555666777')
    cy.get('input[name="firstName"]').type('Sigrid')
    cy.get('input[name="lastName"]').type('Laas-Hurt')
    cy.get('input[name="password"]').type('Password123SLH')
    cy.get('input[name="confirm"]').type('Password123SLH')
    cy.get('#username').type('Something')

    //in order to activate submit button, user has to click somewhere outside the input field
    cy.get('h2').contains('Password').click()
    cy.get('.submit_button').should('be.enabled')
    cy.get('.submit_button').click()

    // Assert that both input and password error messages are not shown
    // next 2 lines check exactly the same, but using different approach
    cy.get('#input_error_message').should('not.be.visible')
    cy.get('#password_error_message').should('have.css', 'display', 'none')

    // Assert that success message is visible
    // next 2 lines check exactly the same, but using different approach
    cy.get('#success_message').should('be.visible')
    cy.get('#success_message').should('have.css', 'display', 'block')
  });
});
```



3. Add a new test when the phone number is not filled in: Add content to the empty test 'User cannot submit data when phone number is absent'

```

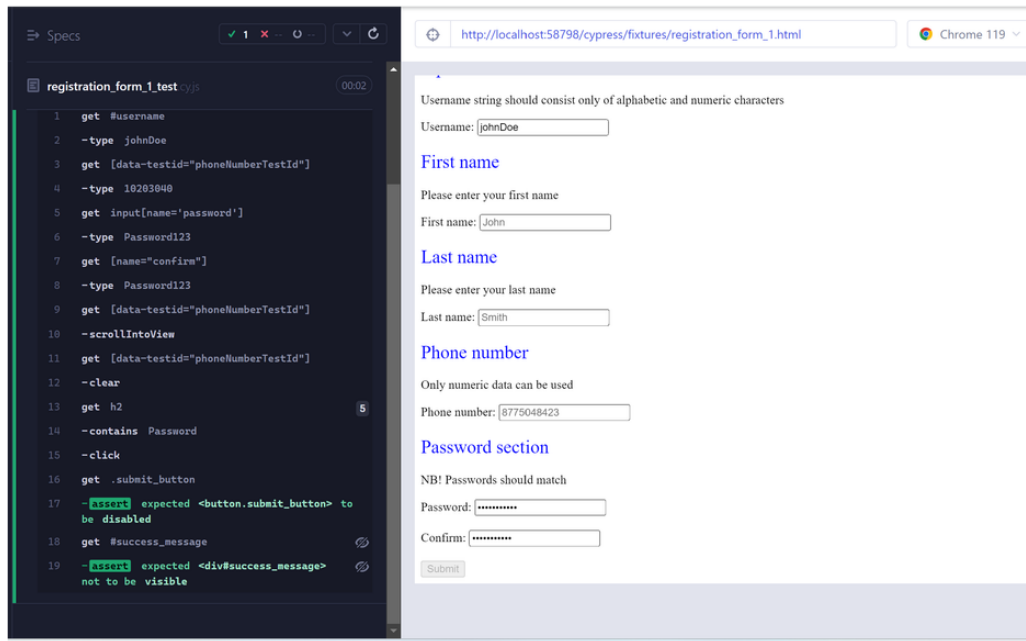
it.only('User cannot submit data when phone number is absent', () => {
  // Add test, similar to previous one with phone number field not filled in
  // All other fields should be entered correctly
  // Assert that submit button is not enabled and that successful message is not visible
  cy.get('#username').type('johnDoe')
  cy.get('[data-testid="phoneNumberTestId"]').type('10203040')
  cy.get("input[name='password']").type('Password123')
  cy.get('[name="confirm"]').type('Password123')

  // Scroll back to phone number input field
  cy.get('[data-testid="phoneNumberTestId"]').scrollIntoView()
  cy.get('[data-testid="phoneNumberTestId"]').clear()
  cy.get('h2').contains('Password').click()

  // Asserting that Submit button is disabled
  cy.get('.submit_button').should('be.disabled')

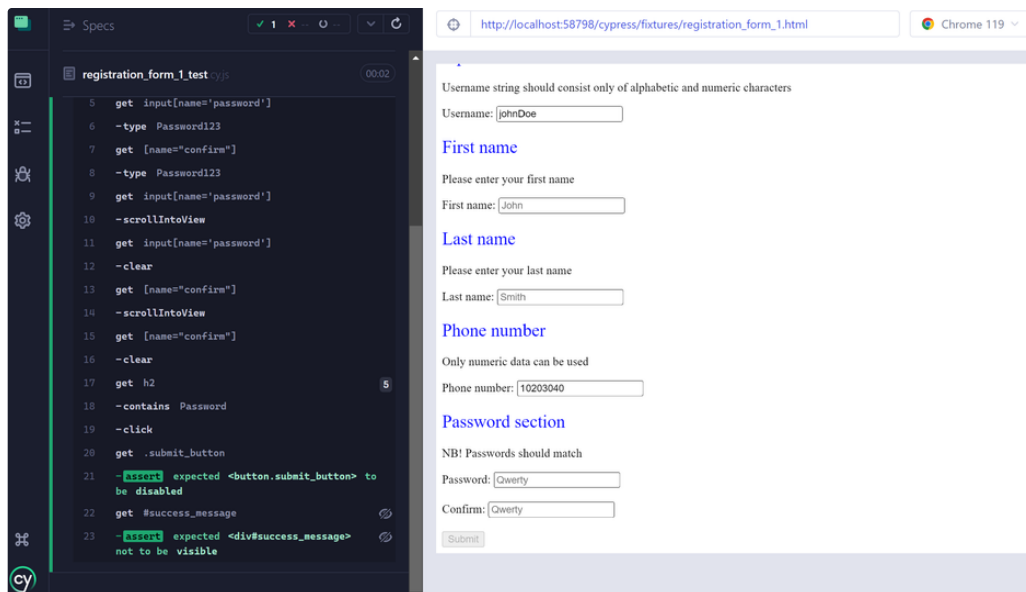
  // Assert that success message is not visible
  cy.get('#success_message').should('not.be.visible')
})

```



4. Add a new test when the password is not filled in: add content to the empty test 'User cannot submit data when password and/or confirmation password is absent'

```
it.only('User cannot submit data when password and/or confirmation password is absent', () => {  
  // Add test, similar to previous one with password field not filled in  
  // All other fields should be entered correctly  
  // Assert that submit button is not enabled and that successful message is not visible  
  cy.get('#username').type('johnDoe')  
  cy.get('[data-testid="phoneNumberTestId"]').type('10203040')  
  cy.get('input[name="password"]').type('Password123')  
  cy.get('input[name="confirm"]').type('Password123')  
  
  // Scroll back to phone number input field  
  cy.get('input[name="password"]').scrollIntoView()  
  cy.get('input[name="password"]').clear()  
  cy.get('input[name="confirm"]').scrollIntoView()  
  cy.get('input[name="confirm"]').clear()  
  cy.get('h2').contains('Password').click()  
  
  // Asserting that Submit button is disabled  
  cy.get('.submit_button').should('be.disabled')  
  
  // Assert that success message is not visible  
  cy.get('#success_message').should('not.be.visible')  
})
```



5. Add a new test that tries to enter letters into the phone number input field (add content to the test 'User cannot add letters to phone number')

```
it.only('User cannot add letters to phone number', () => {
  // Next verification is given as example
  // how we can check from html code, that phone number should contain only numbers
  cy.get('[data-testid="phoneNumberTestId"]').should('have.attr', 'type', 'number')

  // Add steps, when all fields are correctly filled in, except phone number
  // Try typing letters to phone number field
  // Assert that submit button is not enabled and that successful message is not visible
  cy.get('#username').type('johnDoe')
  cy.get('[data-testid="phoneNumberTestId"]').type('10203040')
  cy.get('input[name="password"]').type('Password123')
  cy.get('[name="confirm"]').type('Password123')

  // Scroll back to phone number input field
  cy.get('[data-testid="phoneNumberTestId"]').scrollIntoView()
  cy.get('[data-testid="phoneNumberTestId"]').clear()
  cy.get('[data-testid="phoneNumberTestId"]').type('Cerebrum Hub')
  cy.get('h2').contains('Password').click()

  // Asserting that Submit button is disabled
  cy.get('.submit_button').should('be.disabled')

  // Assert that success message is not visible
  cy.get('#success_message').should('not.be.visible')
})
})
```

Specs

registration_form_1_test cypress 00:02

```
5 get [data-testid="phoneNumberTestId"]
6   -type 10203040
7 get input[name='password']
8   -type Password123
9 get [name='confirm']
10  -type Password123
11 get [data-testid="phoneNumberTestId"]
12  -scrollIntoView
13 get [data-testid="phoneNumberTestId"]
14  -clear
15 get [data-testid="phoneNumberTestId"]
16  -type Cerebrum Hub
17 get h2
18  -contains Password
19  -click
20 get .submit_button
21  -assert expected <button.submit_button> to
   be disabled
22 get #success_message
23  -assert expected <div#success_message>
   not to be visible
```

http://localhost:58798/cypress/fixtures/registration_form_1.html Chrome 119

Username string should consist only of alphabetic and numeric characters

Username:

First name

Please enter your first name

First name:

Last name

Please enter your last name

Last name:

Phone number

Only numeric data can be used

Phone number:

Password section

NB! Passwords should match

Password:

Confirm:

6. Run all the tests from this spec file together

Specs

registration_form_1_test cypress 00:11

This is first test suite, Sigrid Laas-Hurt

- ✓ User can submit data only when valid mandatory values are added
- ✓ User can use only same both first and validation passwords
- ✓ User cannot submit data when username is absent
- ✓ User cannot submit data when phone number is absent
- ✓ User cannot submit data when password and/or confirmation password is absent
- ✓ User cannot add letters to phone number

http://localhost:58798/cypress/fixtures/registration_form_1.html Chrome 119

Username string should consist only of alphabetic and numeric characters

Username:

First name

Please enter your first name

First name:

Last name

Please enter your last name

Last name:

Phone number

Only numeric data can be used

Phone number:

Password section

NB! Passwords should match

Password:

Confirm:

BONUS Assignment 6: FIX TEST FILE WITH FAILED TESTS

Specs

7

registration_form_4_test

00:04

Input fields

✓ Username cannot be empty string

✓ Username tooltip is visible

✓ Username should have min and max length values 1 and 50 characters

✓ Username should support only lower letters and numbers

✓ Email input should support correct pattern

✓ User cannot submit empty registration form

✓ BMW should not be listed in cars list

http://localhost:56480/cypress/fixtures/registration_form_2.html


Chrome 119


Registration form number 2

[Registration form 1](#) [Registration form 3](#)

This is more advanced registration form, some parts are reused from form 1

This is simple registration form





Company Name	Company Description
Company number 1	The best company
Another company nr 2	Competitor of best company

- Item one
- Item two
- Item three
 - Item 3.1
 - Item 3.2

Input username

Username string should consist only of alphabetic and numeric characters

Username:

BONUS Assignment 7: CREATE VISUAL AND FUNCTIONAL TESTS FOR REGISTRATION FORM 3

