

## Task 1: Test Plan

---

- **What are the objectives of the testing in this iteration?**

Answer: The objective of testing in this iteration is to catch faults, errors and failures and create errors so that way we see where the system is flawed or can be improved. As the program is still in the development phase, we should see how each test passes or fails in order to understand what the next iteration should include or not include. The objective is to make sure that all input from the user is reflected onto the console.

I will test the input of the `inputnickname("")` function and the `inputletter("")` function to see if the console logs the output and if it produces bugs or errors. I will in Task 2 check for simple forms of input and output to check that the basics are working and there are no potential bugs there. I also want to test if upper case letters create bugs or produce an error when written as input for these two functions.

- **What to test? Include a short rationale for why you choose to test these objects and not others. (Look in Task 2, 3 for this...)**

Answer: I will be testing if a name is console logged when entered into the `inputnickname("")` function and if singular letters are output in the `inputletter("")` function. I choose these test objects because being able to put a letter and name are the primary options the user has in the beginning of the game, and I want to make sure they do not produce bugs, errors or have faults. I will check that simple alphabetical words in task 2 are correctly logged into these two functions. The rationale for testing these and not others is because the risks of more faults and possible bugs are much greater as the user has the freedom to input more kinds of data into the functions and methods and crashing the game/not being able to play the game correctly. Versus other things ie "tries" which is the amount of time a user can input data. These things have less of a risk of producing flaws as it is a simple algorithm that reduces the number until it reaches zero, and therefore does not have as many variants or abilities to be orthogonal in tests.

### Assignment 3

Then for Task 3 I will use Chai Mocha to test my Javascript code. I decided to tests the inputletter and inputnickname function here too however throw new error functions were implemented to catch several kinds of incorrect inputs that may render the game unplayable. I then also implemented a new “guessword” feature to the game which will fail as the code has been commented out. This guess word has a throw error with message property as well and tests if a user has input more than one word. I want to prevent the user from putting in more than one word as they are only allowed to guess one word for the game of hangman which only includes one word, at least for this game model.

- How this testing is going to be done, what should be dynamically or statically tested, what testing techniques are going to be used. (Look in Task 2, 3 for this...)

Answer: Test 1 and Test 2 can be statically tested, as in reviewed prior to the manual entry tests. The console will ask for an input for nickname for task 1 and for input of a letter in task 2. These tests will be manually done for Task 3 for a dynamic test which will check what errors are “caught” when the wrong value input is created. For Test 1 and Test 2 in Task 3, I will do a system in test that will check if the value is stored and presented correctly. So that will be four tests in total, Test 1.1, 1.2, 2.1, and 2.2.

- Make a time plan for this including time-estimations and measure.

Description	Time plan	Time Actual
Reading instructions/ understanding readings	2hrs	10hrs
Coding updates	1hr	7.5 hrs
Reviewing Code	2 hrs	5 hrs
Manual Test	1hr	2hrs
Restructuring/doing methods	3 hrs	20hrs
Automated Unit code,	2hrs	8hrs

Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

downloading chai and mocha and configuration, test creation		
---	--	--

An approved test-plan must include all of the items above.

---

## Task 2: Manual Test Cases using the client application

---

Test name	Test 1.1	Test 2.1
Use case tested	Name/letter input to output registering	Uppercase letter input in hangman letter
Test Description	inputletter('') function is tested to see if the letter entered is registered in the hangman game	To see if uppercase letters will register as the correct answer when the input == the word array so that way no further if statements or functions are needed to mitigate faults and errors.
Preconditions	Get to html page, open console, press newgame(); then inputletter('') function is typed	Go to html page, open console, enter newgame(); to start.
Test steps	Type in inputletter('d') then enter. Then see if the letter is registered.	Type in uppercase "C" / other letters and see if it registers/ how system reacts
Expected Result	Name is registered and when newgame(); is commanded in console, lower case letters, number of tries, and " there was a 'l' in the word then _ _ _ l _ _ (example) + good job + nickname" will be produced on the screen	If letter is included in the word, it will register in the underscore [] value in lower case. If "A" is used it will be brought back as:  there is a "a" in the word _ _ _ _ _ a _ _ tries: 7 for 'nickname'

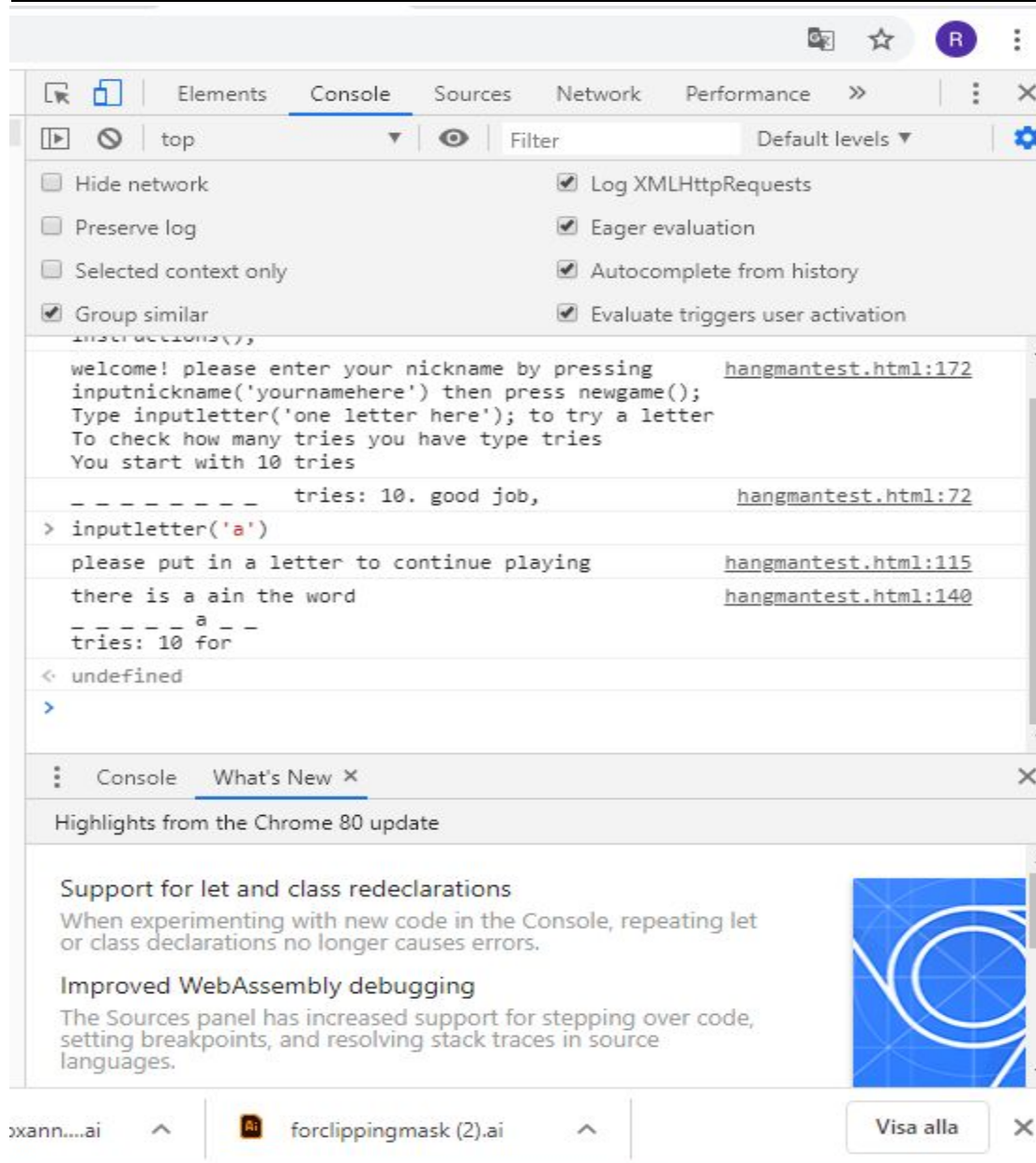
Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

Succeed/Fail?	✓ 1/OK	X 0/F
Final Comments	Letter is repeated back with success.	The .ToLowerCase function is not included here and therefore uppercase letters are not registered in the letter function.

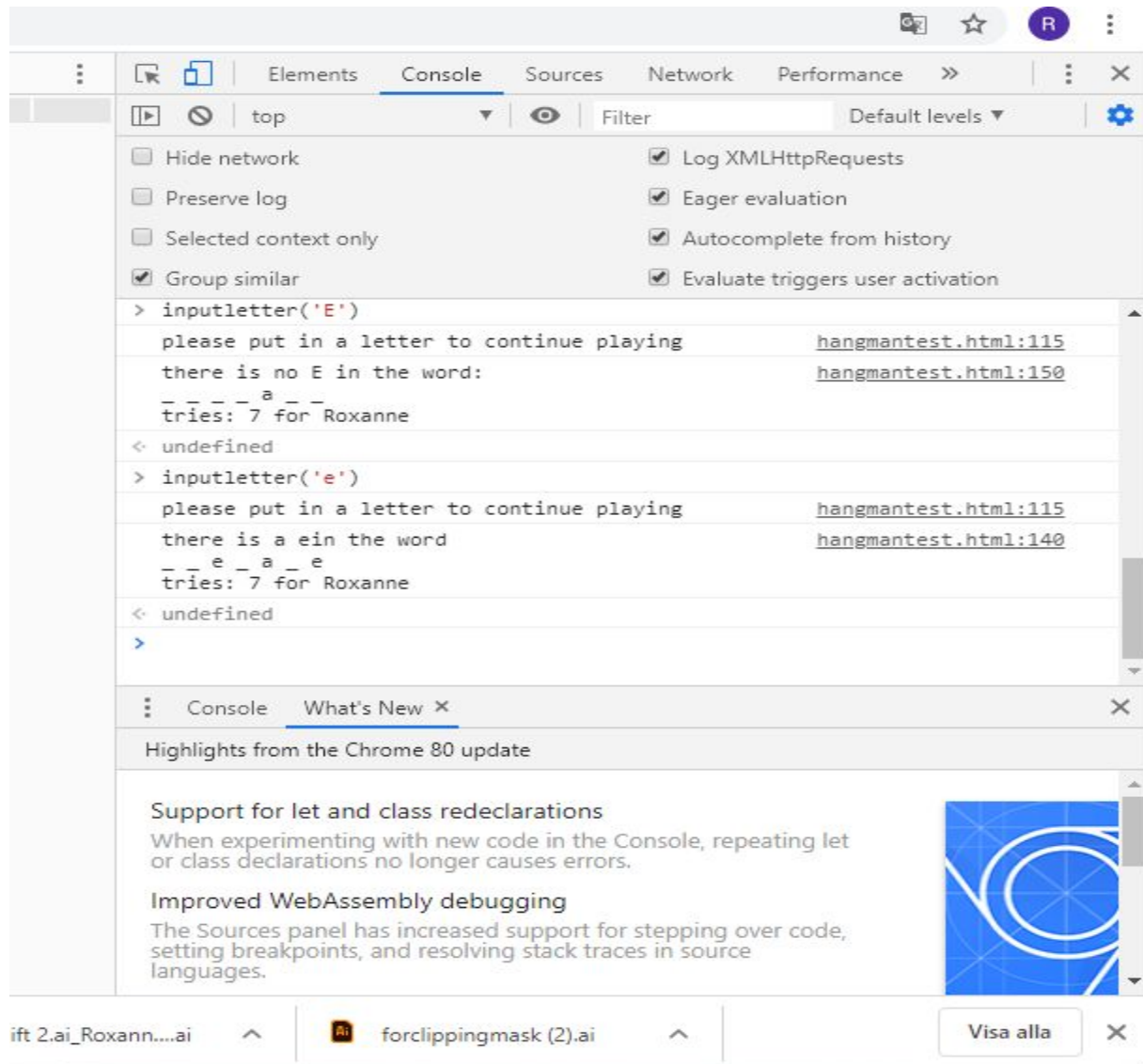


Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3



Test name	Test 1.2	Test 2.2
Use case tested	Name registers in hangman game	Uppercase letter input for nickname function
Test Description	Nickname entered in inputnickname(function) is tested to see if It is produced back in the console when showing nickname/letter in	To see if uppercase letters will register when inputnickname(); function is used and that it is bug and error free so that way nothing has to be caught with an if

Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

	word	statement.
Preconditions	Get to html page, open console, then <code>inputname('')</code> , then <code>newgame()</code> ;	Go to html page, open console, enter <code>inputnickname()</code> ; to start.
Test steps	Type in <code>inputname('roxanne')</code> when name is asked for then, press “enter” key	Type in uppercase letter, ie “Roxanne” / see if it registers as nickname with uppercase letter
Expected Result	Name is registered and when <code>newgame()</code> ; is commanded in console, lower case letters, number of tries, and “good job + nickname” will be produced on the screen	If letter is included in the word, it will register as ‘Roxanne’ with upper and lowercase letters
Succeed/Fail ?	✓1/OK	✓1/OK
Final Comments	Name is repeated back however a bug was caught: an error warning is produced although there is no error	The nickname registered with upper and lowercase letters. Possible because it was an empty array that is getting a string pushed.

Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

The screenshot shows a web browser with three tabs: 'Rebecka Gr...', 'hangman g...', and 'letter to lov...'. The 'hangman g...' tab is active, displaying a Hangman game interface. The console is open, showing the following logs:

```
Hangmangame test game!for instructions type hangmantest.html:177
instructions();

welcome! please enter your nickname by pressing hangmantest.html:172
inputnickname('yournamehere') then press newgame();
Type inputletter('one letter here'); to try a letter
To check how many tries you have type tries
You start with 10 tries

- - - - - tries: 10. good job, hangmantest.html:72
> inputnickname('roxanne')
your name cannot be empty. please enter your name hangmantest.html:101
< 1
> newgame();
- - - - - tries: 10. good job, roxanne hangmantest.html:72
< undefined
```

Below the console, there is a 'What's New' section with the following text:

**Support for let and class redeclarations**  
When experimenting with new code in the Console, repeating let or class declarations no longer causes errors.

**Improved WebAssembly debugging**  
The Sources panel has increased support for stepping over code, setting breakpoints, and resolving stack traces in source languages.

The bottom of the browser shows the taskbar with the name 'Roxann....ai' and a button labeled 'forclippingmask (2).ai'.



Roxanne Albinsson Ileanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

The screenshot shows a web browser with four tabs: 's + mast...', 'Rebecka Gr...', 'hangman g...', and 'letter to lov...'. The 'hangman g...' tab is active, displaying a Hangman game interface. The game text reads: 'welcome! please enter your nickname by pressing inputnickname('yournamehere') then press newgame(); Type inputletter('one letter here'); to try a letter To check how many tries you have type tries You start with 10 tries'. Below this, it says 'tries: 10. good job,'. The user has entered 'Roxanne' as their nickname, and the game responds with 'your name cannot be empty. please enter your name'. The user has also entered '1' as a letter, and the game responds with 'tries: 10. good job, Roxanne'. The user has then entered 'undefined' as a letter, and the game responds with 'undefined'. The browser's developer console is open, showing the game's output and the user's input. The console also displays a 'What's New' section with updates for Chrome 80, including 'Support for let and class redeclarations' and 'Improved WebAssembly debugging'. The bottom of the browser shows two AI-related tabs: 'Uppgift 2.ai\_Roxann....ai' and 'forclippingmask (2).ai', along with a 'Visa alla' button.

Elements Console Sources Network Performance

top Filter Default levels

☐ Hide network ☒ Log XMLHttpRequests

☐ Preserve log ☒ Eager evaluation

☐ Selected context only ☒ Autocomplete from history

☒ Group similar ☒ Evaluate triggers user activation

welcome! please enter your nickname by pressing [hangmantest.html:172](#)  
inputnickname('yournamehere') then press newgame();  
Type inputletter('one letter here'); to try a letter  
To check how many tries you have type tries  
You start with 10 tries

tries: 10. good job, [hangmantest.html:72](#)

> inputnickname('Roxanne')

your name cannot be empty. please enter your name [hangmantest.html:101](#)

< 1

> newgame();

tries: 10. good job, Roxanne [hangmantest.html:72](#)

< undefined

> |

Console What's New

Highlights from the Chrome 80 update

**Support for let and class redeclarations**  
When experimenting with new code in the Console, repeating let or class declarations no longer causes errors.

**Improved WebAssembly debugging**  
The Sources panel has increased support for stepping over code, setting breakpoints, and resolving stack traces in source languages.

Uppgift 2.ai\_Roxann....ai forclippingmask (2).ai Visa alla



Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

Assignment 3

## Task 3: Unit tests

---

I tested `inputnickname("")` with 2 different inputs, one testing a blank array, the other testing one with a length over 15 which should produce an error message/enable the input from going through to the game.

I tested `inputletter("")` with integer input which is caught with a throw error and also tested for more than one character which produces an error. These catches pass the tests in Chai mocha for javascript.

I then created a 3rd method that will be implemented allowing the user to guess a word. When the commented out code is put back into the program, the user should be able to input a guessword which will be presented to them to compare what they have guessed to what the word actually is. The test catches the error of the input being more than one word, although it fails currently because it is commented out. Attached in the assignment is my actual code and tests.

I have taken a screenshot of the actual tests and the tests failing. For reference, I have attached my javascript code/file liabilities in order to get a better picture of what was

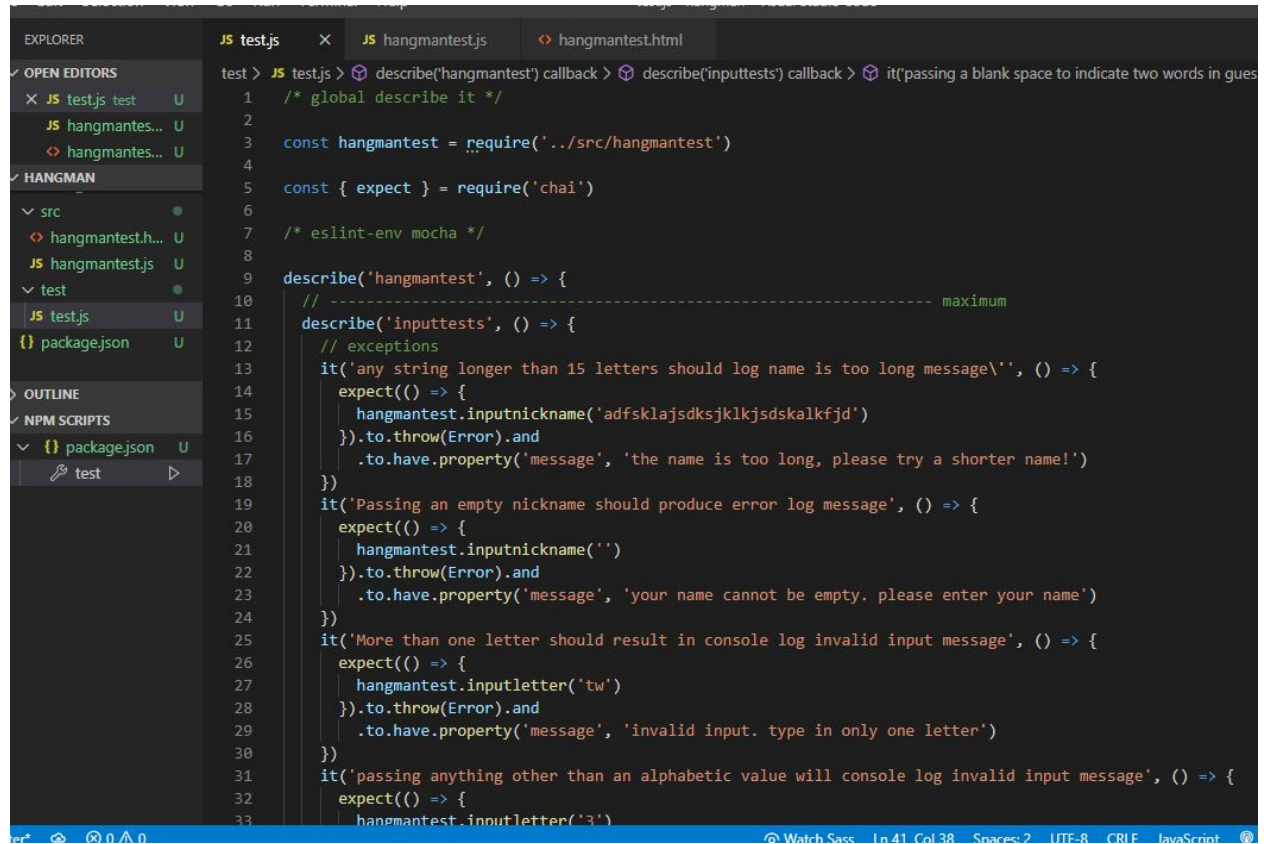
Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

### Assignment 3

tested/how the code is structured.



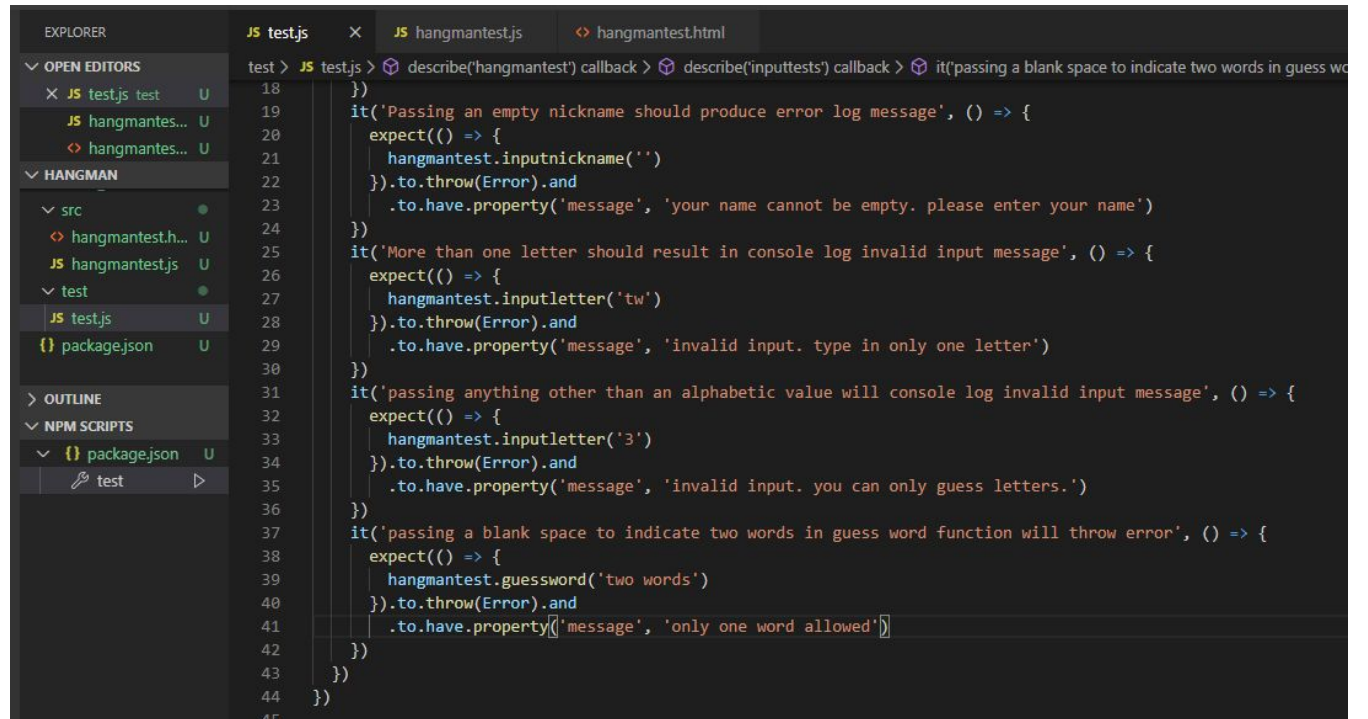
```
test > JS testjs > describe('hangmantest') callback > describe('inputtests') callback > it('passing a blank space to indicate two words in gues
1  /* global describe it */
2
3  const hangmantest = require('../src/hangmantest')
4
5  const { expect } = require('chai')
6
7  /* eslint-env mocha */
8
9  describe('hangmantest', () => {
10    // ----- maximum
11    describe('inputtests', () => {
12      // exceptions
13      it('any string longer than 15 letters should log name is too long message', () => {
14        expect(() => {
15          hangmantest.inputnickname('adfsklajsdksjklkjsdskalkfjd')
16        }).toThrow(Error).and
17          .to.have.property('message', 'the name is too long, please try a shorter name!')
18      })
19      it('Passing an empty nickname should produce error log message', () => {
20        expect(() => {
21          hangmantest.inputnickname('')
22        }).toThrow(Error).and
23          .to.have.property('message', 'your name cannot be empty. please enter your name')
24      })
25      it('More than one letter should result in console log invalid input message', () => {
26        expect(() => {
27          hangmantest.inputletter('tw')
28        }).toThrow(Error).and
29          .to.have.property('message', 'invalid input. type in only one letter')
30      })
31      it('passing anything other than an alphabetic value will console log invalid input message', () => {
32        expect(() => {
33          hangmantest.inputletter('3')
```

Roxanne Albinsson Iheanacho

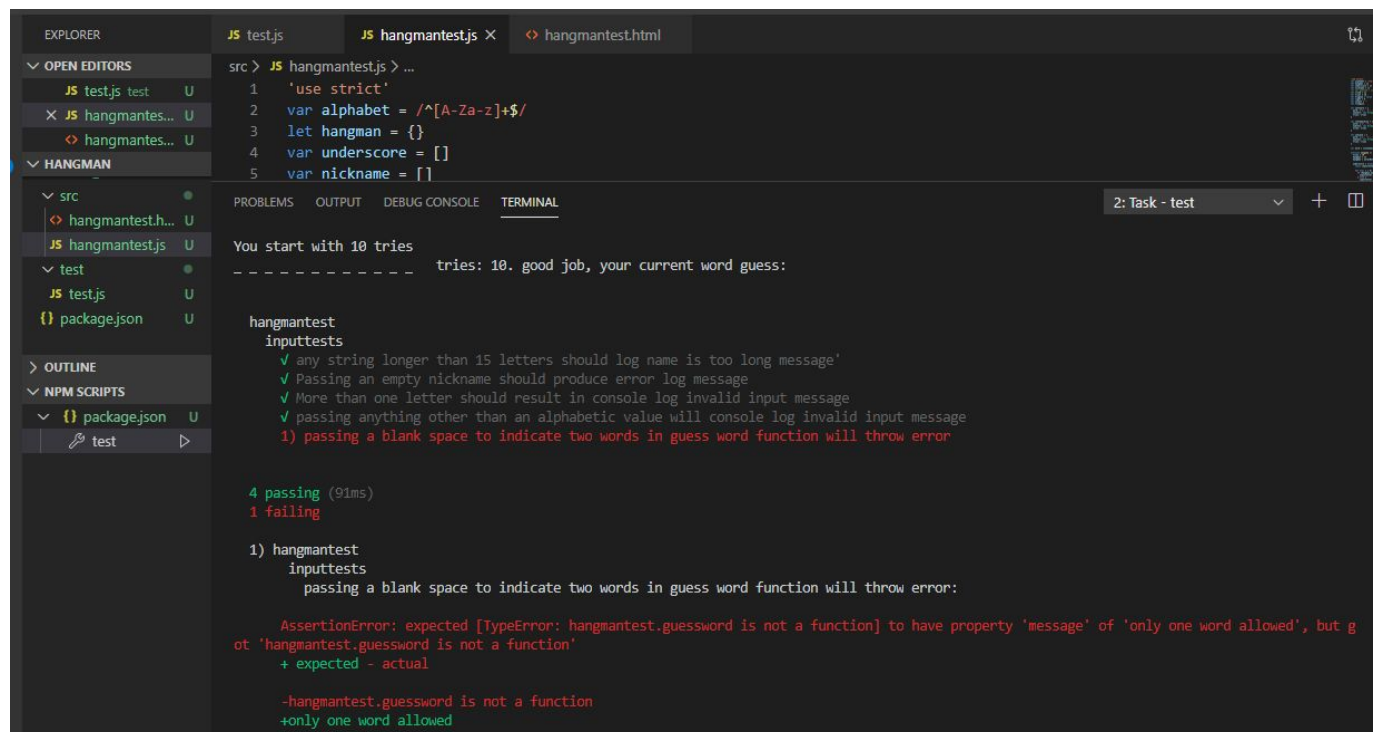
[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

## Assignment 3



```
test > JS testjs > describe('hangmantest') callback > describe('inputtests') callback > it('passing a blank space to indicate two words in guess wo
18 })
19 it('Passing an empty nickname should produce error log message', () => {
20   expect(() => {
21     hangmantest.inputnickname('')
22   }).toThrow(Error).and
23     .to.have.property('message', 'your name cannot be empty. please enter your name')
24 })
25 it('More than one letter should result in console log invalid input message', () => {
26   expect(() => {
27     hangmantest.inputletter('tw')
28   }).toThrow(Error).and
29     .to.have.property('message', 'invalid input. type in only one letter')
30 })
31 it('passing anything other than an alphabetic value will console log invalid input message', () => {
32   expect(() => {
33     hangmantest.inputletter('3')
34   }).toThrow(Error).and
35     .to.have.property('message', 'invalid input. you can only guess letters.')
36 })
37 it('passing a blank space to indicate two words in guess word function will throw error', () => {
38   expect(() => {
39     hangmantest.guessword('two words')
40   }).toThrow(Error).and
41     .to.have.property('message', 'only one word allowed')
42 })
43 })
44 })
45
```



```
src > JS hangmantestjs > ...
1 'use strict'
2 var alphabet = /^[A-Za-z]+$/
3 let hangman = {}
4 var underscore = []
5 var nickname = []

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
2: Task - test

You start with 10 tries
----- tries: 10. good job, your current word guess:

hangmantest
inputtests
✓ any string longer than 15 letters should log name is too long message'
✓ Passing an empty nickname should produce error log message
✓ More than one letter should result in console log invalid input message
✓ passing anything other than an alphabetic value will console log invalid input message
1) passing a blank space to indicate two words in guess word function will throw error

4 passing (91ms)
1 failing

1) hangmantest
   inputtests
     passing a blank space to indicate two words in guess word function will throw error:

    AssertionError: expected [TypeError: hangmantest.guessword is not a function] to have property 'message' of 'only one word allowed', but g
ot 'hangmantest.guessword is not a function'
+ expected - actual

-hangmantest.guessword is not a function
+only one word allowed
```

---

## SHORT CONCLUSION

Roxanne Albinsson Iheanacho

[Ra222tq@student.lnu.se](mailto:Ra222tq@student.lnu.se)

<https://gitlab.lnu.se/1dv600/student/ra222tq/assignment-3/tree/master>

## Assignment 3

---

In conclusion, the test scripts for chai and mocha were helpful in finding bugs. I found them helpful in structuring code and finding the right kinds of methods in order to produce bugs/ catch bugs more clearly. However, something I did not get to test was more deeply ingrained code i.e. my gamewon function. I find the scripts a bit shallow and was not sure how to implement it in further down methods that the user only gets to at the very end of the game. I also found that the tests did not have scripts that could compare the functions I wanted. I wanted to test the `.toLowerCase()` function, but Chai doesn't do that, so I wasn't able to test my code that I wanted to implement because of that. I also realize as in the lectures with Jesper that testing code takes much, much longer than thought and is more complicated than is thought of.