Who:

Ksenia Lepikhina, Wade Myers, Peter Lindee, Andres Barrera, Noel Taterway

Title:

Hooli XYZ

Automated Tests:

Node.js does not have an internal unit or e2e testing framework, so we used two libraries: Chai and Mocha. Chai is just an assertion library to be used with Mocha, which is the actual testing framework. We created three tests, all of which check the basic routes.

The first test, “Login Content”, checks that the route to the login page provides a proper body response. In this case the response from the server should be “Good Response”.
The second test, “Login Status”, checks that routing to the login page returns the correct Status Code (200), which means that the page loaded successfully.
The third automated test, “Main Status”, checks the status code of the index page, and asserts that it should also have successfully loaded. To run the tests, make sure an instance of the server is running in a terminal, and in a separate terminal run “npm test”

Screenshot of test (Provide a copy of the output showing the results of the automated test cases running.)

The following is a screenshot of the three tests:

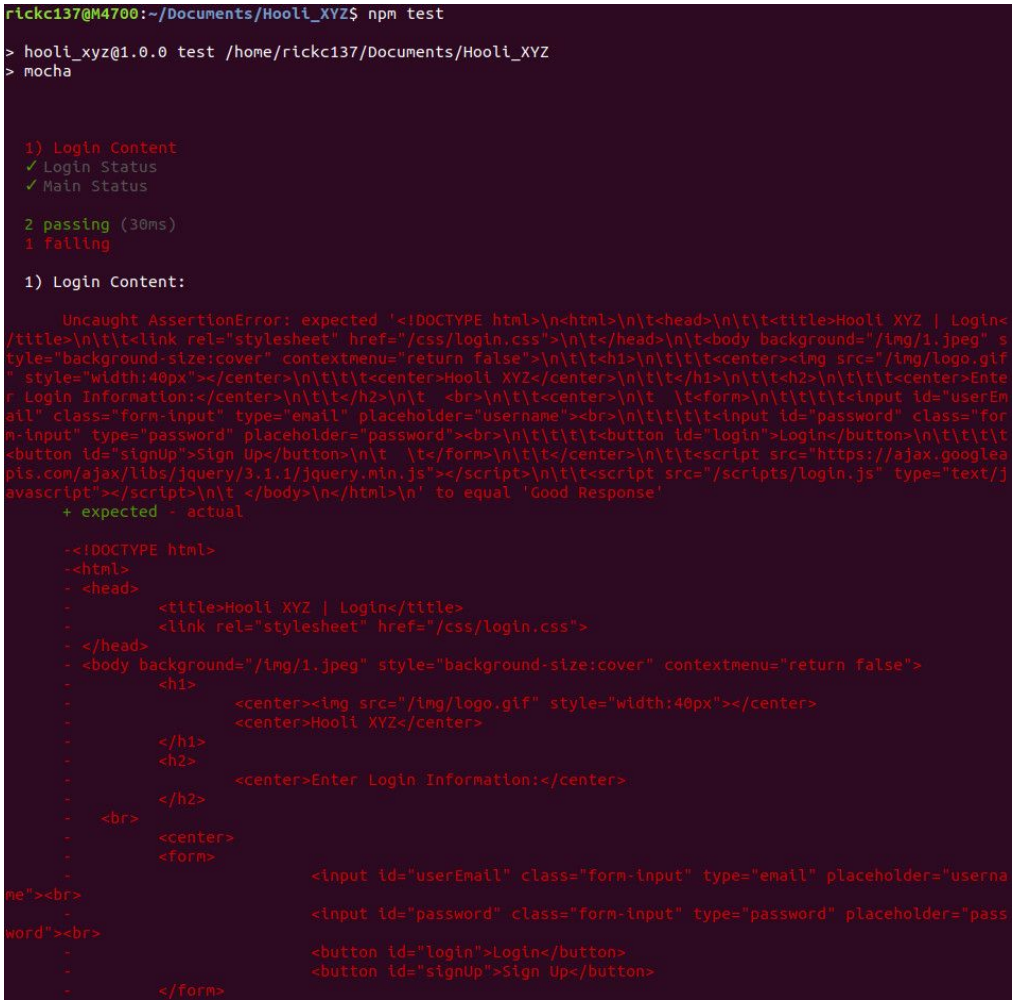
```
const expect = require("chai").expect
const request = require('request');

it("Login Content", function(done){
  request('http://localhost:8001/login', function(err, res, body){
    expect(body).to.equal("Good Response")
    done()
  })
})

it("Login Status", function(done){
  request('http://localhost:8001/login', function(err, res, body){
    expect(res.statusCode).to.equal(200)
    done()
  })
})

it("Main Status", function(done){
  request('http://localhost:8001', function(err, res, body){
    expect(res.statusCode).to.equal(200)
    done()
  })
})
```

The following is a screenshot of the results of the automated test cases:



Explain what the test is and what it does

Did it fail or pass

As of right now, two of the tests pass, and one fail. Both the status code tests (2 and 3) pass because the server is correctly navigating to those routes. The first test fails because it is outdated. The test expects the string “Good Response” but is actually getting the html for the Login Page (as it should). All in all the tests return what they should.

User Acceptance Tests:

Story 1:

Who:

Who created test cases: Ksenia Lepikhina

Test Date:

06/28/2018

Software Version:

Browser: Google Chrome Version 67.0.3396.99

Database: MySQL Ver 8.0.11

Operating System: macOS High Sierra (version 10.13.5)

Prerequisites:

Needs to be run on a PC with a browser

Needs to have MySQL installed

Needs to be able to start a node server with “npm start”.

Purpose:

Verify: “As a CU student, I want a website where I can store my files.”

Test	Expected Results	Actual Results	Reqs Validated	Pass/Fail
A user should be able to see their files when they enter a folder	See files	See files	yes	pass
A user should be able to upload folders/files	Working upload button	Upload button that does not upload a file	no	fail
A user should be able to see the files they have uploaded	See uploaded file	No file uploaded	no	fail
A user should be able to download files	Working download button	Download button that does not download	no	fail

Story 2:

Who:

Who created test cases: Ksenia Lepikhina

Test Date:

06/28/2018

Software Version:

Browser: Google Chrome Version 67.0.3396.99

Database: MySQL Ver 8.0.11

Operating System: macOS High Sierra (version 10.13.5)

Prerequisites:

Needs to be run on a PC with a browser.

Needs to have MySQL installed.

Needs to be able to start a node server with “npm start”.

Purpose:

Verify: “As a CU student, I want my files and folders to be secure.”

Test	Expected Results	Actual Results	Reqs Validated	Pass/Fail
A user should be able to login with a valid username and password	Able to enter text, Submit, and see folders page	Able to enter text, Submit, and see files page	yes	pass
A user should be able to see JUST their folders when they login	Just see folders	See everyones files	no	fail
A user should be able to see JUST the files they have uploaded	See uploaded file	Not implemented	no	fail

Story 3:

Who:

Who created test cases: Luke Myers

Test Date:
06/28/2018
Software Version:
Browser: Google Chrome (v8)
Database: N/A
Operating System: Ubuntu 17.10
Prerequisites:
Needs to be run on a PC with a browser that can render HTML and CSS.
Needs to be able to start a node server with “npm start”.
Purpose:
Verify: “As a CU student, I want the site to be aesthetically pleasing.”

Test	Expected Results	Actual Results	Reqs Validated	Pass/Fail
Be able to load front page image	CSS needs to display	CSS displays	yes	pass
Be able to load logo image	See logo on front page	See logo on front page	yes	pass
Be able to see CSS on file page	See CSS on file page	See CSS on file page	yes	pass
If page fails to load	Throw error	Not implemented	no	fail