



ncstate-csc-coursework / csc326-2025-fall-tp-204-5



&lt;&gt; Code

Issues 33

Pull requests

Actions

Projects 1

Wiki

Security

Edit

New issue



# Team A code review #112

Open



mdeisenb\_ncstate opened 1 hour ago · edited by mdeisenb\_ncstate

Edits ▾ ⋮

Looking at frontend files: ([#91](#)) App.jsx, AdminHomeComponent.jsx, AppSidebarLayout.jsx

For backend, looking at: ([#93](#))  
JwtTokenProviderTest.java & jwtAuthenticaitonTest.java.

Create sub-issue ▾



mdeisenb\_ncstate 42 minutes ago · edited by mdeisenb\_ncstate

Edits ▾ Author ⋮

## Going through Guiding Quesitons...

### Code inspection of Bennett's & Will's code from team A, iteration 1B

- This inspection looked at the pull requests with numbers [Cbzug/uc1 implementation #91](#) & [Cbzug/uc1 implementation #93](#) for the backend & frontend of UC1, which is for user login.

### Code Inspection Checklist

Is the code Javadoced with meaningful comments that help the reader understand the method?

- The javadocs are good for Bennett's documents. Could use a little more description in the frontend.

Are the comments up to date with the current implementation of the method?

- Yes, they are up to date with the current implementation.

Are there ways to simplify the implementation or reduce redundancy?

- I wouldn't say so. The tests in the backend are nice and concise. The frontend does its job well of providing a good looking interface.

Is the functionality at the right location in the architecture or should it be moved to a different layer?

- Yes, the functionality is at the right location.

Is there appropriate input validation? (Just b/c the front end has a check for invalid inputs, doesn't mean there isn't a way to get invalid input to the backend.)

- The front end does not have checks for invalid inputs, but that should be handled in the backend.

Does the implementation meet the requirements?

- The implementation doesn't just meet the requirements but exceeds them. The frontend looks amazing Bennett!

Is there missing functionality?

- Does not seem so.

Are there error paths that aren't handled?

- They should be handled in the backend.

Are the variable names meaningful?

- Yes, they are meaningful.

## Test Inspection Checklist

---

Are the test ids or test methods unique?

- The test methods are indeed unique.

Are descriptions, repeatable, specific, and unambiguous?

- The descriptions make sense for UC#1.

Are expected results clearly stated?

- The expected results are clearly stated.

Are preconditions defined for each test?

- Yes, preconditions are defined for each test.

Is the test behavior tied to a requirement?

- Yes, the test behavior is tied to a requirement.

Does each test accurately verify a specific requirement?

- Each test does accurately verify a specific requirement.

Does the test evaluate the right results to know that it passed?

- The test does evaluate the right results to know it passed.

Do the tests sufficiently cover the requirements? Are more tests needed?

- 3 test cases is enough for the first use case.

## Overall Summary of Bennett's Code

- Bennett's and Will's work on the frontend & backend is overall great. An overall great looking interface has been created because of his work. Only thing that could be added is more comments for the frontend code. The backend does a good job of verifying correct tokens as well!



mdeisenb\_ncstate 5 minutes ago

Author



### Frontend Files

#### App.jsx

- Great that you removed the components from the remnants of coffeemaker for the imports.
- Good use of and tags to allow for authenticated users to view other parts of the homepage.
- Overall, code explains itself relatively well and looks good.
- Could use some more comments.

#### AdminHomeComponent.jsx

- Looks like the frontend stores the username of the admin. Good since no GET request will be needed whenever loading the homepage since it will display the message "Hello {adminName}"
- Overall nice and concise.
- Would like some comments detailing lines 7-14.

#### AppSidebarLayout.jsx

- Good use of imports from lucide-react.

- Lines 146-202 do feel a little hard to read and understand. Perhaps if I used react more often, I would better to be able to understand.
- Would like a comment section explaining the sidebarcontent tag pair at 146 & 179.



mdeisenb\_ncstate now

Author ...

## Backend Files

### JwtAuthenticationFilterTest.java

- I can see that user jdoe is the one that is tested for authentication.
- Initial impression is that there is a lack of comments in the file.
- Would like an explanation in the comments on how the mock requests are done.

### AppSidebarLayout.jsx

- I like the setup method for each test. Would be good however to explain what you're doing, especially in regards as to what line 30 and 31 do.
- The test names are concise and self explanatory.
- Would like to see what happens when a token is generated for a user that doesn't exist.



Add a comment

Write

Preview

H B I | ☰ <> 🔗 | ☰ ☰ ☰ | @ 🗨️ ↩️ 📎

Use Markdown to format your comment

Paste, drop, or click to add files

✓ Close issue



Comment

## Metadata

Assignees



No one - [Assign yourself](#)

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

Code with agent mode

[Create a branch](#) for this issue or link a pull request.

Notifications

Unsubscribe

You're receiving notifications because you're subscribed to this thread.

Participants



- Transfer issue
- Duplicate issue
- Lock conversation
- Pin issue

