TOP JS-EXPRESS ISSUES

- Following instructions
- Lack of 'refresh'
- Not understanding multiple users
- Same old things
- Unclean PRs
- Separation, but not of Concerns
- Comments
- CSS Specificity

FOLLOWING INSTRUCTIONS

I provide instructions in the README

- Good - If you read them and follow them
- Good - If you read them and ask for clarification
- Not Good - If you completely ignore them

If you do an assignment and it technically "works", but you didn't follow instructions so you end up not using/showing the intended skill, what grade do you deserve?

MISSED INSTRUCTIONS

- *The default message/users should be removed*

- *Will send their user name as a hidden field (name=username) to the server when posting a new message*

- *You must use the route paths as given/described*

CONSEQUENCES

The instructions not only cover what is worth points

It is also the point of the exercise

Failure to follow instructions can be REALLY PAINFUL

LACK OF "REFRESH"

- *Will have a "Refresh" button that loads the base chat page*

BUT WHY?

Reasonable question!

To update the message list without having to send a message

Other users may be posting

MULTIPLE USERS

Many students tried to save `curUser` (why not `currentUser`?!) just like the user list or message list.

Those are true for all users, while `curUser` *isn't*

- User A POSTS to `/login`, sets `curUser`, redirects to `/`
- User B POSTS to `/login`, sets `curUser`, redirects to `/`
- User A GETS `/`, uses `curUser`, gets page for User B

Hard to do manually; very likely with multiple users

Instead, redirect to `/?username=UserB`, don't save `curUser` at all

SAME OLD THINGS

- Don't Use `var`
- Variable names
- Always use strict comparison (===)
  - Not loose comparison (==)
  - Unless you use truthy/falsy
- Use truthy/falsy where you can
- Understand when to use an array vs object
- CSS(HTML) class names
  - Don't embed the tag names
  - Describe the data, not what it looks like

LOTS OF UNCLEAN PRS

- Merge conflicts
- Inappropriate files

MERGE CONFLICTS

Each assignment is in it's own directory

There should be no conflicts if you only have files from this assignment

You can have multiple PRs out at a time - order doesn't matter

HOW IS ALL OF THIS HAPPENING

Most likely: You are skipping a key step

```
git checkout -b basic-js
(do work)
(add/commit/push)
     # time passes
git checkout master   # Key Step!
git pull origin master
git checkout -b js-express
# New Branch is based off of the branch you __were in__
( do work, etc )
```

CHECK!

You have access to the PR

- You can see what it is "in" it
- You can fix it before it is even created
- You can fix it after it is created before being told

Experts still make plenty of mistakes

- they just check and fix them unprompted
    - Which teaches them to not make them as often

NOT-QUITE SOC

- chat.js - data (Model)
- chat-web.js - HTML (View)
- server.js - routes (Controller)

Separate so each can change with minimal knowledge of another

- Allows for multiple views, models, etc
- Imagine a desktop app, or mobile app, etc

Where does a `login.js` or `login-web.js` fit in? What about shared bits?

Not **wrong**, but probably more cost than benefit

REMEMBER TO UPDATE COMMENTS

For too many

```
const sender = 'Bao'; // Hardcode until we add a login
```

Became

```
const sender = req.body.username; // Hardcode until we add a login
```

Do you see the problem?

Inaccurate or misleading comments **worse** than no comment

DEBUGGING REPORTS

`console.log()` during development is normal and fine

`console.log()` to generate useful output is fine (on server)

- There are better tools/modules, but the idea remains fine

`console.log()` for debugging in submitted code is not fine

COMMENTED OUT CODE

Commenting out code during development is normal and fine

Submitting commented out code is not fine

- Why is it there?
- How long will it be there?

CSS OVER-SPECIFIC

Compare

```
.message {
  border: 1px solid black;
}
```

To

```
div.message {
  border: 1px solid black;
}
```

Or even

```
li > div.message {
  border: 1px solid black;
}
```

Be only as specific as you actually want to say