

# Reviewing Project1 Lessons

Key lessons:

- You cannot trust user data
  - cookies, url, params all user data
- MVC is easiest solution to flow
  - Update state data
  - Generate View based on data
- Software is hard to estimate!
  - Never gets easier
  - Final will not have extensions

# **You cannot trust user data**

- It CAN be wrong
  - Mistakes
  - App changes
  - Deliberate

# **User Data comes in many forms**

- Which url are they going to?
  - The order you intend can be bypassed
- Cookies & session id
  - May not be accurate
- Passed parameters
  - May not be the options you offered
  - Must allowlist

# **Easy to assume, but worth it not to**

- We assume the intended/offered

But the issue isn't just "security"

- Defensive coding re: user data
  - Also better code in general
  - Easier to change
    - Fewer built in assumptions

# **Web apps rapidly become complex**

- Deal with most recent data
- Deal with data spanning multiple requests

How do we keep the code clean?

- Understandable
- Easy to change

# Separate the state and the view

Remember the MVC Pattern?

- Good lessons on separating concerns
  - Even when not explicitly using MVC

Have a clear data model (state)

- Based on request data
  - Update state
- Render new view
  - Based on state

# When writing a route handler

Ask yourself

- "What is the user asking to do?"
- "What state changes does this make?"

Write your code to answer those questions

THEN:

- Write returned view
  - View may have conditionals on state
  - View should never CHANGE state

# **Actual general programming advice**

None of those points are specific to web

- General best practices
- Web just forces you to handle earlier
  - Client-Server
  - Stateless request-response
  - Multiple users asynchronously requesting
  - Poor separation of concerns hits you faster



# **Did Project1 take longer to write than expected?**

## Important Lesson

- Software always more time than you think
- Because it is all about abstraction
  - Describing a lot of work in a quick summary
- Never gets easier

# Remember this lesson for future

- Bosses/Clients/Team want info ASAP
- You will be asked for timelines
  - ALWAYS allow extra time!
    - Not lying, you *know* your gut is wrong
  - Avoid guessing; Break problems down
  - More on this at end of semester
- This was even worse than you think
  - compare code was pre-written
  - Login code was pre-written

# **Final Project has no extensions**

I have a grading deadline that can't be pushed back

- Final project is custom
  - No pre-written parts
- Final project has more parts
  - REST Services
  - React code
  - CSS

Don't treat project like a weekly assignment

- Start early, alongside assignments