React Development & Debugging

A New Workflow: Test-Driven Debugging

1. Implement New Feature
   a. Plan with Copilot
   b. Get Copilot to write the code
   c. Get Copilot to fix typescript errors
   d. Run existing tests and resolve any issues
      - No need to make tests yet
2. Try the New Feature
   a. Use the app manually
   b. Notice issues (styling, content, interactivity)
   c. Create automated tests for them, then fix them

Code Writing Guidance for the Copilot

- I've created a `react_practices.md` file with suggested practices to follow
- When asking copilot to write code, remind it of these practices by dragging file into the chat and telling it to follow them.
    - Especially remind when you are having discussions about how to organize code for a new feature
- Once in a while ask it to review the code so far with respect to the practices in that document.
- After you've been using React for a week or so, go back and look at the document and try to understand everything it is saying. Ask the copilot to explain things to you.
    - You can try now, but it might be easier to understand after just a bit of experience.

Chrome DevTools

Key Tools For Us:

- Elements panel: Right-click element -> Inspect
- Console panel: Error messages; Log messages

Other Useful Panels (eventually):

- Sources panel: View source files, set breakpoints
- Network panel: Monitor network requests
- React Developer Tools (separate extension)

Similar things available in other browsers.

- I encourage you to use Chrome for web development.
- I'll show things in Chrome dev tools

Live Coding Example

- I made a degraded version of our app with some bugs and missing features
- We'll live code to fix them, following these practices

If you want to follow along:

- `git fetch upstream`
- `git checkout -b upstream/week6-session11-react-debugging`
- `npm run dev`

Debugging Styling

- Look at the page to see if it "looks right"
- If not:
    i. Check if the styles you expect are actually applied
    ii. If not, debug style application
    iii. If so, debug .css style contents

Debug .css Style Contents

- Take a screenshot of how it's displaying and of the .css styles applied to it
- Paste into CoPilot
- Explain what you wanted it to look like
- Ask it to propose alternative ways to achieve that
- Discuss and then pick one to try
- Repeat as needed
- If it seems like you're not making progress
    - Ask copilot to stop making changes and explain what things it is trying and why

Debug Style Application

- Common display issue is that style you think is applied is actually not.
- Select an element in the Elements panel
- Look at the Styles panel to see what styles are applied
- If the style you expect is not there, create a test:
  - Take a screenshot of styles panel and paste into CoPilot
  - Explain what style you expected to be applied to that element
  - Ask it to create an automated test to check that the style is applied
- Then try to fix it
  - Ask copilot to generate possible explanations for why the style is not applied
  - Ask it to verify which of those explanations is correct, and only then try to fix

Debug Presence or Absence of DOM Elements

- Check whether the DOM contains the right elements
- If missing expected element, or have an extra:
  - Ask copilot to create a test to check for the presence/absence of the element
  - Ask it to propose possible explanations for why the element is missing/extra
  - Ask it to verify which of those explanations is correct, and only then try to fix
  - See next slide for common explanations and how to verify them

Debug Interactivity

Common explanations for wrong content after a user interaction (e.g., button click, form submission):

- Wrong event handler fires
- State doesn't update as expected
- Rendered output doesn't match state/props as expected

How to verify such explanations:

- Ask copilot to add logging to event handlers, state updates, and render methods
- Cut and paste console output into copilot and ask it to interpret it

Resources

- React Docs: Debugging
- React Developer Tools
- Common React Bugs

Questions?