

Class 10: Mock Technical Interview Preparation

1. Review: Last Class
2. Mock Technical Interview Preparation
3. Studio - Homework 3, Sprint 2

Copyright 2026, Kyle J. Harms

Review: Progressive Web App

A Progressive Web App (PWA) is a type of web application that is installable on a user's device and can work offline or with a poor network connection.

PWA requirements:

- A manifest (provides metadata about the app)
- A service worker (enables offline functionality)
- HTTPS (for security)

Next Week

Plan: Next Week

Today: Oral Exam I Preparation

This plan is tentative and subject to change.

- **Monday, 9/28:** Oral Exam I Begins + Homework 3 due
- **Tuesday, 9/29:** LLM Prompt Engineering
- **Thursday, 10/1:** Libraries/Packages & LLM APIs
- **Friday, 10/2:** Oral Exam I Ends

Homework 4 released sometime this week (depending on progress with LLMs).

Oral Exam I

A Mock Technical Interview

Oral Exam I: A Mock Technical Interview

Learning Objective: Demonstrate your ability to quickly prototype components and communicate your understanding of technical concepts in a professional setting.

Employers are seeking candidates with **power** skills, like communication and teamwork, in addition to technical skills. This oral exam is designed to help you practice communicating your design and development process for prototyping apps.

Mock Technical Interview Format

Think: A causal (but important) conversation with a possible employer (course assistant).

You will be provided an exam repository with a partially implemented app.

You will be asked to implement a component in the provided codebase when given a task from a GitHub issue. You will have a maximum of **15 minutes** to complete the task and answer any questions about your implementation.

Think aloud as you work through the task to explain your thought process and reasoning.

You may use Vue.js, Mozilla, TailwindCSS, and DaisyUI documentation. No other reference resources, like notes, are permitted; **you may not use AI, including GitHub Copilot.**

Mock Technical Interview Grading

Pass	Criteria
You got the job! (full credit)	The answer clearly communicates your understanding the content.
You're not their top candidate (half credit)	The answer is missing some important details, but overall demonstrates some understanding of the content.
You didn't get the job (no credit)	The answer does not demonstrate sufficient understanding of the content.

Mock Interview Preparation

Strategy:

1. Study the existing code and the task first. (Don't just start typing code in a panic.)
2. Identify what you have and what you need.
3. Think through the methods you know and how they can be used to provide what you need.
4. Solve **one** problem at a time. Break bigger problems into smaller problems. (Don't try to solve everything at once.)
5. Incrementally start coding. Code one thing at a time and test it.
6. **Explain** what you are doing and **why** you are doing it. (Think aloud.)

Activity: Mock Interview Preparation

Prepare:

- Open practice exam repository as a Codespace.
- Review the codebase.
- Clear your head. Take a few deep breaths. Relax. You got this.

Exam Practice: With a peer take turns mock interviewing each other.

Interviewer: If the interviewee isn't thinking aloud, ask them to explain their thought process and why they are doing what they are doing.

Interviewee: Think aloud and explain your reasoning as you work through the problem.

Oral Exam I: Scheduling

Schedule your interview time slot in Canvas no *later* than Sunday, Sept 27th, 11:59pm.

The oral exam time slots are on a first-come, first-served basis. If you do not sign up for an oral exam time slot by the deadline, you will receive a 0 for the exam.

Not showing up to your scheduled appointment or canceling within 24 hours of your scheduled appointment will result in 0 credit for the exam. No exceptions.

Activity: More Mock Interview Preparation

Pair up with a classmate and develop several “GitHub issues” to **add a component** to an existing interface prototype. (See back of handout.) Then do it!

Issues should require demonstrating the following skills:

- creating components
- passing props
- using reactive variables
- binding props to reactive variables
- handling events
- emitting events
- conditionally rendering using `v-if`
- conditionally styling elements using `:class` bindings
- using the `v-for` directive to render lists of elements
- employing watchers to react to changes in reactive variables
- binding inputs to reactive variables with `v-model`

Studio: Homework 3, Sprint 2

Goal: Create an installable app prototype that documents the functionality of your UI component library.

Sub-Goals: Implement progressive web app (PWA) functionality in your Homework 3 prototype app.

Sub-Goals: Implement additional components that are necessary to implement an “image conversion” app.

What's Next

Sunday, 9/27: Sign up for Oral Exam I time slot in Canvas

Monday, 9/28: Oral Exam I begins + Homework 3 due

Tuesday, 9/29: LLM Prompt Engineering

Friday, 10/2: Oral Exam I ends