Web3, Session 12

Exam

Format

- Oral exam
- 20 minutes including setup, deliberation, and all that
- You will draw to cards with numbers 1 - 6
- The numbers correspond to the 6 assignments
- You choose one of the two assignments to begin with
- In the next ~7 minutes
  - You show and explain some of your code
  - I ask questions of your code
  - I ask general questions of the relevant theory (details later)
- We then proceed with the other assignment

Example

- You draw numbers 1 and 2
- You handed in the TypeScript/Vue.js/GraphQL assignments (1-3) as one project
- We will cover the TypeScript and Vue.js aspects of the project and the theory
- Maybe you choose to start with Vue.js
- You show the code
- I will ask about e.g. binding, control structures, re-rendering, components with slots, routing, props and emits and state management
- Don't prepare a presentation
- It's okay to have notes, but remember that we can see if you're reading

What's relevant for each assignments

- Assignment 1
  - TypeScript types and how you use them to avoid errors
- Assignment 2
  - Vue.js and state management
- Assignment 3
  - GraphQL theory and implementation
- Assignment 4
  - Functional programming and tools to help with functional programming
- Assignment 5
  - Redux, RxJS
- Assignment 6
  - SSR, Next.js

Assignment 1

- From course description:
  - Explain the elements of the TypeScript type system
  - Explain the function of TypeScript utility types
  - Apply OO programming in TypeScript
  - Design and implement a web application using one or more of the techniques taught in the course
- I will ask about e.g. basic type operators, discriminating unions, casting and narrowing, immutability, utility types, type manipulations

Assignment 2

- From the course description:
  - explain relevant client programming design patterns.
  - explain mechanisms for rendering and re-rendering
  - apply at least two web client frameworks and at least two state management frameworks
  - argue for the choice of state management techniques in web client
  - design and implement a web application using one or more of the techniques taught in the course
- I will ask about e.g. binding, control structures, re-rendering, components with slots, routing, props and emits and state management

Assignment 3

- From the course description
  - implement a server using TypeScript
  - design and implement a web application using one or more of the techniques and technologies taught in the course
- I will ask about e.g. GraphQL types and queries, resolvers, GraphQL client, web sockets, server-sent events, GraphQL subscriptions.

Assignment 4

- From the course description:
  - explain the definition of functional programming
  - apply functional programming in TypeScript
  - apply callbacks and higher-order functions in TypeScript
  - design and implement a web application using one or more of the techniques and technologies taught in the course
- I will be asking about e.g. immutability, purity, functional libraries, pipelining (with map, filter, flatMap, etc.), currying, persistent data structures (immutable.js), lightweight proxies (immutable.js sequences)

Assignment 5

- From the course description:
  - explain relevant client programming design patterns
  - explain mechanisms for rendering and re-rendering
  - explain the components of reactive programming
  - apply at least two web client frameworks and at least two state management frameworks
  - use reactive programming in the client
  - design and implement a web application using one or more of the techniques and technologies taught in the course
- I will be asking about e.g.: One-way data flow, reducers, slices, thunks, (RxJS) observables, subjects, pipes and operators, merge vs concat

Assignment 6

- From the course description:
  - implement server-side rendering
  - argue for the choice of server-side and client-side rendering
  - design and implement a web application using one or more of the techniques and technologies taught in the course
- I will be asking about e.g. static vs dynamic pages, client vs server components, app routing, hydration, how to avoid hydration errors, state management in SSR applications

Questions?