TW-008 TEAM LEAD VERSION (Sprint-6 Week-1)

Meeting Agenda

▶ Icebreaking

▶ Questions

▶ Interview Questions

▶ Coding Challenge

▶ Video of the week

▶ Retro meeting

▶ Case study / project

Teamwork Schedule

Ice-breaking 5m

- Personal Questions (Study Environment, Kids etc.)
- Any challenges (Classes, Coding, studying, etc.)
- Ask how they're studying, give personal advice.
- Remind that practice makes perfect.

Team work 5m

- Ask what exactly each student does for the team, if they know each other, if they care for each other, if they follow and talk with each other etc.

Ask Questions 15m

1. If you see the following import in a file, what is being used for state management in the component?

`import React, {useState} from 'react';`

A. stateful components
B.math
C. React Hooks
D. class components

Answer: C

2. If a function component should always render the same way given the same props, what is a simple performance optimization available for it?

A. Wrap it in the `React.memo` higher-order component.
B. Implement the `useReducer` Hook.
C. Implement the `useRouter` Hook.
D. Implement the `shouldComponentUpdate` lifecycle method.

Answer: A

3. What can you use to handle code splitting?

A. `React.memo`
B. `React.split`
C. `React.lazy`
D. `React.fallback`

Answer: C

4. What is styled-component styling in React?

A. These styles are written as attributes and are passed to the element.
B. It is a JavaScript library for styling React applications. It removes the mapping between styles and components, and lets you write actual CSS augmented with JavaScript.
C. It is basically a .css file that is compiled. When compiled, it produces two outputs. One is CSS that is a modified version of input CSS with the renamed class names. The other is a JavaScript object that maps the original CSS name with the renamed name.
D. It offers a different approach in which no CSS needs to be written to style an application. Instead, It uses utility classes for each CSS property that you can use directly in your HTML or JSX.

Answer: B

5. What is `[e.target.id]` called in the following code snippet?

```
handleChange(e) {
  this.setState({ [e.target.id]: e.target.value })
}
```

A.a computed property name
B.a set value
C.a dynamic key
D.a JSX code string

Answer: C

6. What is sent to an `Array.map()` function?

A.a callback function that is called once for each element in the array
B.the name of another array to iterate over
C.the number of times you want to call the function
D.a string describing what the function should do

Answer: A

7. What do you need to change about this code to get it to run?

```
class clock extends React.Component {
  render() {
    return <h1>Look at the time: {this.props.time}</h1>;
  }
}
```

A.Add quotes around the return value
B.Remove `this`
C.Remove the render method
D.Capitalize `clock`

Answer: D **Explanation:** In JSX, lower-case tag names are considered to be HTML tags. Read this article

8. How do you invoke setDone only when component mounts, using hooks?

```
function MyComponent(props) {
  const [done, setDone] = useState(false);

  return <h1>Done: {done}</h1>;
}
```

A. `useEffect(() => { setDone(true); });`
B. `useEffect(() => { setDone(true); }, []);`
C. `useEffect(() => { setDone(true); }, [setDone]);`
D. `useEffect(() => { setDone(true); }, [done, setDone]);`

Answer: B

9. You have created a new method in a class component called handleClick, but it is not working. Which code is missing?

```
class Button extends React.Component{

  constructor(props) {
    super(props);
    // Missing line
  }

  handleClick() {...}
}
```

A. `this.handleClick.bind(this);`
B. `props.bind(handleClick);`
C. `this.handleClick.bind();`
D. `this.handleClick = this.handleClick.bind(this);` *Answer: D*

10. What property do you need to add to the Suspense component in order to display a spinner or loading state?

```
function MyComponent() {
  return (
    <Suspense>
      <div>
        <Message />
      </div>
    </Suspense>
  );
}
```

A. lazy
B. loading
C. fallback
D. spinner

Answer: c

11. Using object literal enhancement, you can put values back into an object. When you log person to the console, what is the output?

```
const name = 'Rachel';
const age = 31;
const person = { name, age };
console.log(person);
```

A. `{{name: "Rachel", age: 31}}`
B. `{name: "Rachel", age: 31}`
C. `{person: "Rachel", person: 31}}`
D. `{person: {name: "Rachel", age: 31}}`

Answer: B

12. What is `setCount`?

```
const [count, setCount] = useState(0);
```

A. the initial state value
B. a variable
C. a state object
D. a function to update the state

Answer: D

Interview Questions 15m

1. What is the difference between Element and Component?

Answer: An Element is a plain object describing what you want to appear on the screen in terms of the DOM nodes or other components. Elements can contain other Elements in their props. Creating a React element is cheap. Once an element is created, it is never mutated. Whereas a component can be declared in several different ways. It can be a class with a render() method. Alternatively, in simple cases, it can be defined as a function.

2. What is the difference between state and props?

Answer: Both props and state are plain JavaScript objects. While both of them hold information that influences the output of render, they are different in their functionality with respect to component. Props get passed to the component similar to function parameters whereas state is managed within the component similar to variables declared within a function.

3. What is the difference between Shadow DOM and Virtual DOM?

Answer: The Shadow DOM is a browser technology designed primarily for scoping variables and CSS in web components. The Virtual DOM is a concept implemented by libraries in JavaScript on top of browser APIs.

4. What are the lifecycle methods of React? Answer:

Before React 16.3
componentWillMount: Executed before rendering and is used for App level configuration in your root component.
componentDidMount: Executed after first rendering and here all AJAX requests, DOM or state updates, and set up event listeners should occur.
componentWillReceiveProps: Executed when particular prop updates to trigger state transitions.
shouldComponentUpdate: Determines if the component will be updated or not. By default it returns true. If you are sure that the component doesn't need to render after state or props are updated, you can return false value. It is a great place to improve performance as it allows you to prevent a re-render if component receives new prop.
componentWillUpdate: Executed before re-rendering the component when there are props & state changes confirmed by shouldComponentUpdate() which returns true.
componentDidUpdate: Mostly it is used to update the DOM in response to prop or state changes.
componentWillUnmount: It will be used to cancel any outgoing network requests, or remove all event listeners associated with the component.

React 16.3+
getDerivedStateFromProps: Invoked right before calling render() and is invoked on every render. This exists for rare use cases where you need derived state. Worth reading if you need derived state.
componentDidMount: Executed after first rendering and here all AJAX requests, DOM or state updates, and set

up event listeners should occur.
shouldComponentUpdate: Determines if the component will be updated or not. By default it returns true. If you are sure that the component doesn't need to render after state or props are updated, you can return false value. It is a great place to improve performance as it allows you to prevent a re-render if component receives new prop.
getSnapshotBeforeUpdate: Executed right before rendered output is committed to the DOM. Any value returned by this will be passed into componentDidUpdate(). This is useful to capture information from the DOM i.e. scroll position.
componentDidUpdate: Mostly it is used to update the DOM in response to prop or state changes. This will not fire if shouldComponentUpdate() returns false.
componentWillUnmount: It will be used to cancel any outgoing network requests, or remove all event listeners associated with the component.

5. How often does the React useState update? Why?

Answer: Since developers use `useState` to enhance performance by creating queues, React doesn't update changes immediately. Candidates should know that useState doesn't implement changes to the state object directly; instead, the updates occur asynchronously.

Coding Challenge 20m

- Coding Challenge: Random User App

☕

Coffee Break 10m

☕

Videos of the Week 5m

- React Router Setup in 5 minutes

Retro Meeting on a personal and team level 5m

Ask the questions below:

- What went well?
- What went wrong?
- What is the improvement areas?

Case study/Project 15m

Case study should be explained to the students during the weekly meeting and has to be completed in one week by the students. Students should work in small teams to complete the case study.

- RP-03 Task Tracker App

Closing 5m

-Next week’s plan

-QA Session
