💻React + JavaScript Interview Tayyorlash Hujjati

✂️ `var`, `let`, `const` farqi - var: function scope, hoisting mavjud, qiymatini o'zgartirish mumkin - let: block scope, hoisting bor, lekin TDZ (Temporal Dead Zone) bilan, qiymatini o'zgartirish mumkin - const: block scope, qiymati o'zgarmaydi (primitive), hoisting bor, TDZ mavjud

✒️React nima va nima uchun kerak - React – JavaScript kutubxona (library) - UI yaratish uchun ishlatiladi, SPA yaratishga yordam beradi - Virtual DOM orqali UI tez yangilanadi - JSX bilan HTML va JS bir joyda ishlaydi

🛏️JSX nima - JSX – JavaScript XML, React sintaksis kengaytmasi - HTML va JS kodini bir joyda yozish imkonini beradi - Brauzer JSX'ni to'g'ridan-to'g'ri tushunmaydi, Babel orqali JS ga aylantiriladi

🌂 State nima va props bilan farqi - State: komponent ichidagi mahalliy ma'lumotlar, faqat o'sha komponentni yangilaydi, useState orqali ishlatiladi - Props: tashqi komponentdan ma'lumot olish, read-only - State o'zgarganda komponent render bo'ladi

🌡️ Functional vs Class Components - Class: ES6 klass, render() metodi, lifecycle methodlari (componentDidMount, ...), this.state bilan state - Functional: oddiy funksiya, return bilan JSX, useState, useEffect bilan state va lifecycle - Zamonaviy Reactda functional componentlar afzal

🎒Lifecycle vs useEffect - Class: componentDidMount, componentDidUpdate, componentWillUnmount - Functional: useEffect hook bilan lifecycle ishlari bajariladi

🎓React Hooks - useState – state yaratish - useEffect – lifecycle ishlari - useContext – global state olish - useRef – DOM elementlarga referens yaratish

🎩Conditional Rendering

```
{isLoggedIn ? <Dashboard /> : <Login />}
```

- Shartga qarab komponent render qilinadi

🎽List rendering

```
{items.map(item => <li key={item.id}>{item.name}</li>)}
```

- key unik bo'lishi kerak

🕘Event Handling

```
<button onClick={() => setCount(count + 1)}>Click</button>
```

- onClick, onChange va boshqalar - this functional componentlarda yo'q, classda bind qilish kerak

✂️✂️React Context - Global state yaratish uchun - Props drillingdan qochish imkonini beradi

✂️✒️React Router - SPA navigatsiyasi uchun - BrowserRouter, Route, Link komponentlari bilan ishlaydi

✂️🛏️Performance Optimization - React.memo – componentni unnecessary renderdan saqlaydi - useMemo, useCallback – hisoblashlarni optimallashtiradi

✂️🌂Closures (JS)

```
function outer() {
  let count = 0;
  return function inner() {
    count++;
    return count;
  }
}
```

- Inner function outer scope'dagi o'zgaruvchilarni eslaydi

✂️🌡️ Async/Await

```
async function fetchData() {
  const res = await fetch('/api');
  const data = await res.json();
}
```

- Promise bilan ishlashni osonlashtiradi

✂️🎒Array Methods - map, filter, reduce, find, some, every - Array metodlari bilan data transformatsiyasi va filtering

✂️🎓 Event Bubbling va Capturing - Bubbling: child → parent elementga event yuqoriga tarqaladi - Capturing: parent → child elementga event pastga tushadi

✂️🎩Pure vs Impure Functions (React Performance) - Pure: bir xil input bilan har doim bir xil output beradi, side effect yo'q - Impure: side effect bor, output inputga bog'liq emas

✂️🎽 Key React Concepts - Virtual DOM – tez UI yangilanishi - Reconciliation – React o'zgargan qismlarni DOMga yangilaydi - Component Composition – komponentlarni bir-biriga joylash

✒️🏮 Controlled vs Uncontrolled Components (Form Handling) - Controlled: state orqali inputni boshqaradi - Uncontrolled: ref orqali DOMdan value oladi