

Del 1 - endast begrepp och koncept

1. Ramverket React (en översikt, vad som är unikt för ramverket, fördelar, ev nackdelar, osv)

React, ett ramverk, är en JavaScript-bibliotek som används för att bygga interaktiva och effektiva användargränssnitt för både webbplatser och applikationer.

En fördelen med React är att man kan bygga återanvändbara UI-komponenter. Dessa komponenter kan kombineras för att skapa komplexa användargränssnitt och gör utvecklingen mer modulär.

En annan fördel med React och som gör den unik är dess virtuella DOM (Document Object Model) där den används som virtuell representation av webbsidan för att uppdatera endast de nödvändiga delarna, vilket i sin tur förbättrar prestandan.

En nackdel med React är inlärningskurvan. Det tar lite tid att lära sig och konfigurera. Det krävs även noggrann optimering av koden för att bibehålla optimal prestanda då React-applikationer blir mer komplexa kan storleken på den bundlade koden öka och påverka prestanda och laddningstider.

2. Vad innebär Rendering och Virtual DOM?

Rendering är processen där webbsidans innehåll, inklusive text, bilder och andra element, visas för användaren i webbläsaren. Inom React sker renderingen med hjälp av komponenter.

Virtual DOM är en virtuell representation av webbsidan och används av React för att effektivt hantera uppdateringar av webbsidan. React jämför den virtuella DOM:en med den faktiska DOM:en och uppdaterar endast de delar som har förändrats.

3. Vad är JSX? Vad används det till?

JSX (JavaScript XML) låter utvecklare skriva HTML-liknande kod direkt i sina JavaScript-filer. Istället för att skapa och manipulera DOM-element genom JavaScript-metoder kan JSX användas för att deklarera komponenter med HTML-liknande syntax.

4. Vad är ett undantag inom programmering? Eng. Exception

Ett undantag inom programmering innebär en oväntad händelse eller ett fel som inträffar under programkörningen. Det normala utförandeflödet avbryts och programmet hoppar till en speciell kodavsnitt som hanterar felet.

5. Vad innebär autentisering inom webbapplikationer? Vad används det till?

Autentisering inom webbapplikationer är processen av att verifiera och bekräfta användarens identitet för att ge användaren åtkomst till specifika resurser eller funktioner. Det används för att säkerställa att endast auktoriserade användare kan komma åt skyddad information eller utföra vissa handlingar på en webbplats eller i en applikation.

Del 2 - sammanfattning av projekt

För att bygga min portföljsida använde jag olika komponenter för att organisera och återanvända min UI-kod. Exempelvis har jag en "Header"-komponent för att kunna visa en gemensam rubrik överst på varje sida, en "NavBar"-komponent för att visa navigeringslänkar och en "Projects"-komponent för att visa information om ett projekt.

Genom att använda Reacts rendering och "reconciliation" kunde jag göra effektiva uppdateringar av mitt användargränssnitt. När en av mina komponenter uppdateras jämför React den tidigare och nuvarande versionen av den virtuella DOM:en för att identifiera förändringar. Genom att bara uppdatera de nödvändiga delarna av DOM:en undvek jag onödiga renderingar och förbättrade prestandan för min portföljsida. En annan fördel var att det var väldigt tidseffektivt att använda komponenter för sedan även börja Tailwind som gjorde det hela mycket enklare och intuitivt.

Genom att använda JSX kunde jag skriva en läsbar och strukturerad kod för att beskriva hur mina UI-element skulle se ut och interagera med varandra. Min JSX-kod transpilerades sedan till JavaScript-funktionsanrop som React kunde använda.

När jag skriver en return-sats inuti en komponent kallas det som finns inuti return-satsen för JSX (JavaScript XML). Genom att använda JSX kunde jag skriva en läsbar och strukturerad kod för att beskriva hur mina UI-element skulle se ut och interagera med varandra.

Inuti return-satsen kan jag använda HTML-liknande taggar för att skapa element som divar, knappar, formulär och textfält. Jag kan också inkludera JavaScript-uttryck och använda mig av variabler för att skapa dynamiska UI-element. JSX gör det enklare att kombinera och sammanfoga olika UI-komponenter och skapa en hierarki av element som representerar strukturen i mina användargränssnitt. JSX-koden transpileras sedan till JavaScript-funktioner som används av React för att skapa och uppdatera komponenter.