

React opfrisker

Teoretiske spørgsmål

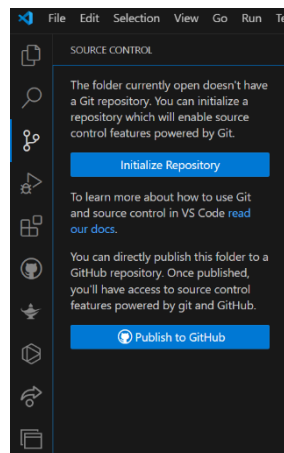
Spørgsmålene besvares skriftligt

1. **Hvad er React og hvad bruges det til?** Det er et brugervenligt Javascript bibliotek der bruges til at udvikle i frontend.
2. **Hvad er en komponent i React?** Komponenter gør koden meget mere overskuelig og nem at vedligeholde. Og de kan nemt genanvendes.
3. **Hvilke forskellige metoder kan du bruge til at style dine React-komponenter? Diskuter fordele og ulemper ved mindst to metoder.** Den nemmeste og mest overskuelige måde er at have en style mappe med css filer der linker til de individuelle komponenter. CSS-moduler giver mulighed for lokaliserede stilarter ved at generere unikke klasse-/id-navne. Men det kræver en opsætning, der understøtter CSS-moduler.
4. **Hvad er state i React?** Et objekt, der repræsenterer komponentens dynamiske data og bruges til at styre komponentens interaktive dele. State giver komponenter mulighed for at reagere på brugerinput, netværksanmodninger og andre begivenheder ved at opdatere komponentens rendering i realtid.
5. **Hvad er props i React?** Props bruges til at sende data fra en forælderkomponent til en børnekomponent. Dette gør det muligt for komponenter at modtage input, der kan ændre, hvordan de opfører sig eller vises.
6. **Hvordan adskiller state sig fra props?** State er lokal for komponenten og kan kun ændres internt, mens props gives til komponenter af deres forældre og er uforanderlige inden for komponenten.
7. **Hvad er hooks i React?** Hooks er funktioner, der giver dig mulighed for at "hooke ind i" Reacts funktionalitet i funktionskomponenter.
8. **Forklar, hvordan useState og useEffect hooks fungerer med eksempler. useState:** Giver dig mulighed for at tilføje state til en funktionskomponent. Det giver dig mulighed for at gemme og opdatere værdier i en komponent på en enkel måde.
useEffect: bruges til at udføre sideeffekter i funktionskomponenter. Dette kan være alt fra at hente data til at tilmelde sig og afmelde sig event listeners.
9. **Hvad er JSX?** Det bruges til at beskrive hvordan brugergrænseflade skal se ud. Det gør det muligt at skrive HTML-lignende kode inde i JavaScript, hvilket gør det nemmere at opbygge og forstå strukturen af komponenter.
10. **Hvordan adskiller JSX sig fra almindelig JavaScript?** JSX er en syntaks, der ligner HTML, og bruges i React til at skrive komponenter i JavaScript, men det skal omdannes til regulært JavaScript, fordi det ikke er gyldigt JavaScript i sig selv.
11. **Hvad er console.log() og hvilken rolle spiller det i JavaScript udvikling?** en metode i JavaScript, der bruges til at udskrive oplysninger til webbrowserens konsol. Denne metode er en af de mest anvendte værktøjer til debugging og overvågning af kode under udviklingen.
12. **Hvornår er det nyttigt at bruge console.log()?** Det kan bruges til fejlfinding og forståelse af kode.

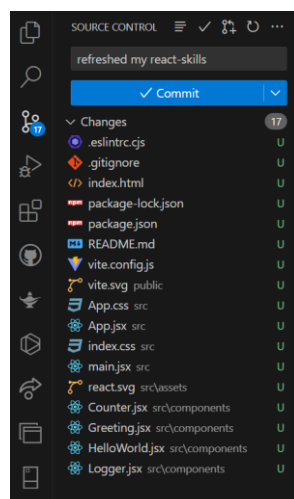
Praktiske opgaver

1. Opret et nyt vite-projekt med et selvvalgt navn
2. Opret en komponent der returnerer en hilsen som f.eks. "Hello, World!"
3. Opret en komponent, der tager 'name' som en prop og viser en besked som "Hello, [name]!".
4. Lav en komponent, der bruger useState hook'en til at tælle, hvor mange gange en knap er blevet trykket på.
5. Lav en komponent, der bruger useEffect til at logge en besked til konsollen, hver gang komponenten renderes.
6. **Tilføj din skriftlige besvarelse som en PDF i roden af projektet.**

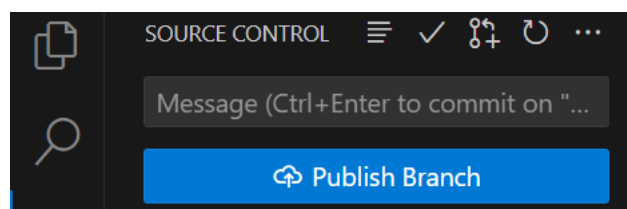
7. Åbn Source Control og tryk på initialize repository for at oprette et nyt repository på Github.



8. Skriv en commit-besked i 'message'-feltet og tryk 'commit'.



9. Tryk derefter 'Publish Branch' og vælg 'Public repository'.



10. Send linket til dit repository til mig på Teams når du er færdig