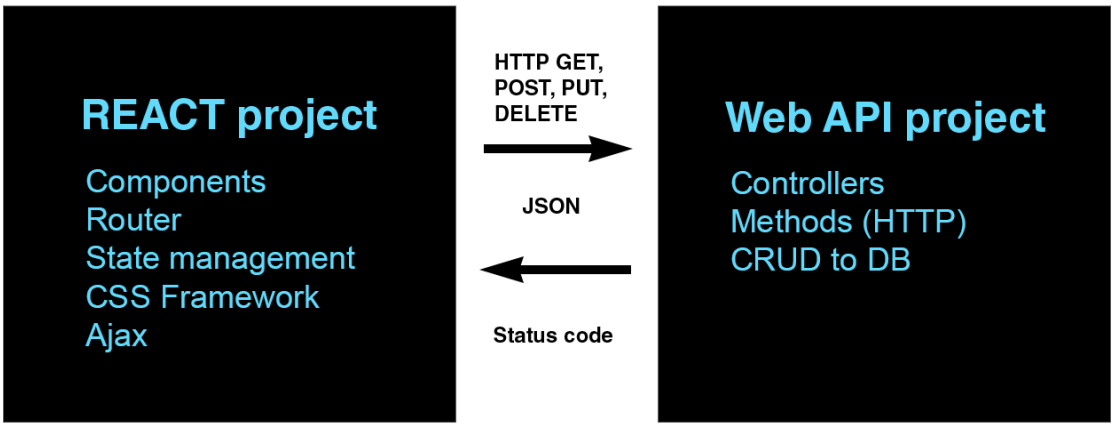
PGR305 Webutvikling 3

Eksamen H2021

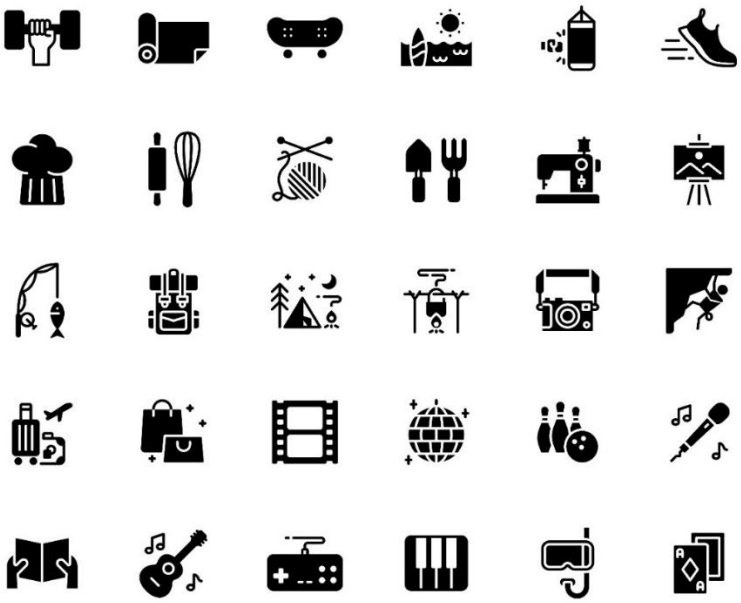
Diverse informasjon:

- Dette er en hjemmeeksamen
- Kan løses alene eller i gruppe på opptil 3
- Hovedtemaene for denne eksamenen er React (frontend) og .NET/C# Web Api med MongoDb (backend)
- Inkluder hele React-prosjektet i din besvarelse, men uten node_modules-mappen
- Legg React-prosjektet og .NET/C#-prosjektet i en felles hovedmappe som zippes før opplastning til WISEFlow
- Husk at oppkoblingen til MongoDb krever en ConnectionString med brukernavn og passord. Du må vurdere om denne burde fjernes slik at sensorene evt. ikke skal kunne se sensitive data. M.a.o. det er ok å levere uten ConnectionString
- Ikke inkluder innloggingsfunksjonalitet som krever at en må taste inn riktig brukernavn og passord for å komme inn i løsningen ved kjøring; dette kan lede til at sensorene ikke får evaluert deler av besvarelsen.
- Du kan fritt bruke bilder fra nettet i denne besvarelsen siden det er en lukket eksamen. Samme gjelder tekster fra nettet.



Skjermbilde over: illustrerer hovedmomentene ved eksamenen

Case: Fullstack applikasjon rundt et tema



I denne eksamenen skal du ta utgangspunkt i ett tema (i listen i neste setning) og utvikle en fullstack løsning rundt informasjon om temaet. Velg én av følgende temaer:

- Ett eller annet med klær, vesker, klokker e.l.
- Pop-artister / musikere
- Fotballspillere / fotballag
- MMA-utøvere
- Om en historisk hendelse mellom 1600-tallet og 1800-tallet
- Én spesifikk TV-serie

Du skal utvikle full CRUD-funksjonalitet i denne sammenheng som betyr at brukere skal kunne lese informasjon gjennom nettsidene, legge til, oppdatere og slette informasjon.

Nedenfor følger de store linjene for frontend og backend, men oppgaven er ganske åpen og du har i stor grad frihet til å utforme løsningen etter egen smak. For både karakterens del og læringens del: Ikke tenk minimalistisk! Mer omfang og mer kompleksitet = bra!

Backend:

- Database i MongoDb Atlas
- .NET/C# Web API med CRUD-funksjonalitet mot MongoDb

Frontend:

- Sider med funksjonalitet hvor en bruker kan gjøre CRUD. Forsøk å dele opp CRUD-funksjonaliteten i flere komponenter.
- Diverse funksjonalitet som du mener kan passe å ha med. Eksempler kan være å ha med søkefunksjonalitet, sorteringsfunksjonalitet, quiz osv. Du har stor frihet til å bestemme funksjonalitet du kan ha med her.

Tips til utviklingen

- Begynn med å planlegge hvilken informasjon du ønsker å jobbe med. Dvs. hvilke properties ønsker du skal være med? Det kan også være greit å i et dokument prøve å skissere alle teknikkene og alle bestanddeler i din applikasjon for å gi deg selv bedre mental oversikt over arbeidet du må gjøre; dette før du begynner å kode noe.
- Gjør gjerne preliminære enkle tester for å sjekke at du får til kobling mellom Web Api og MongoDb og utvid deretter.
- Siden frontend er avhengig av backend kan det være lurt å først få på plass MongoDb, så noen få metoder i Web Api som kan testes mot, og så gradvis bygge ut frontend, og deretter frem og tilbake med utvidelser. Uansett kan man alltids utvide på alle tre steder og det kan være greit å utvide litt og litt istedenfor å ha altfor mange properties på «objektene» til å begynne med som øker sjansen for å gjøre feil og gjør det vanskeligere å finne feil.
- Merk at det er lov å (i god tid før levering) konferere med lærer om løsningen.

Detaljer rundt forventede teknikker

Følgende lister nedenfor er for å gi en oversikt, og sjekkliste, for hvilke teknikker/teknologier som er forventet i denne eksamenen.

Frontend

- Komponentbasert utvikling med blant annet List og Item-fordeling av ansvar
- State management med Context
- TypeScript
    - Interface
    - Type
    - Typesetting
- Router
- Ajax
- React Bootstrap
    - Et hovedmoment her er å gjøre bruk av Grid-systemet i CSS-rammeverket
- Diverse JS-teknikker som for eksempel .map(), spread operator, ternary operator, array, object literal osv.

Backend

- .NET/C# Web Api
- Controller
- Model-klasser
- Interface
- Service
- Diverse konfigurering som for eksempel for CORS
- MongoDb Atlas
- CRUD – Create, Read, Update, Delete
- Bildeopplasting

Sensorveiledning for studenter og sensorer

Følgende tabell gir retningslinje for hvordan eksamenen skal karaktersettes. Merk at tallene er ca.-tall med tanke på at man kan utvide koden sin mer i frontend og/eller backend på forskjellige måter.

Teknikk	Prosentangivelse
FRONTEND med React (ca. 50%)	
Komponentbasert utvikling med blant annet List og Item-fordeling av ansvar. TypeScript med Interface og Type og diverse typedefinering. Diverse JS-teknikker som for eksempel .map(), spread operator, ternary operator, array, object literal, Ajax osv.	Ca. 30-40%
State management med Context. Routing.	Ca. 10-15%
React Bootstrap	Ca. 10-15%
BACKEND med .NET/C# Web Api (ca. 50%)	
Interfaces, Model-klasser, Service-klasser med CRUD og Controllere	Ca. 30-45%
Diverse konfigurasjon, CORS, staticFiles osv-	Ca. 5-10%
Omfang (mengde) og kompleksitet (hvor vanskelig er teknikken), og ryddighet og struktur er også med å styre % man får per del.	

--- Eksamenstekst ferdig ---