Projektuppgift i JavaScript

I denna uppgift ska du arbeta med ett eget projekt byggt helt i JavaScript.
Idén för projektet väljer du själv. Om du saknar en egen idé finns en lista med förslag längre ned.

Målet är att visa att du självständigt kan ta dina kunskaper och gå från idé- och konceptfas till en färdig applikation.

Projektet ska använda ett RESTful API i Node.js med JWT-baserad autentisering och MongoDB som dokumentdatabas.

Viktigt!

Om du väljer att inte göra en traditionell webbapplikation (t.ex. en inbäddad applikation) måste du omedelbart diskutera omfattningen med din lärare. I sådana fall kan testkraven för applikationen skilja sig.

Kravspecifikation

Icke-funktionella krav

- Backend ska tillhandahålla ett RESTful API som kan användas med valfri frontend.
- Applikationen ska fungera i alla moderna webbläsare.
- Applikationen ska vara responsiv och fungera på alla skärmstorlekar.

Funktionella krav (obligatoriska)

- En användare måste kunna registrera ett konto.
- En användare måste kunna logga in.
- En användare måste kunna söka i applikationen.
- En administratör måste kunna logga in på en enkel dashboard och skapa/uppdatera/radera användare.

Funktionella krav (frivilliga – för högre betyg)

- Administratör bör kunna tilldela behörigheter baserat på roller.
- Administratör bör kunna skapa/läsa/uppdatera/radera roller.
- Administratör bör kunna skicka e-post från dashboarden.

Designkrav

Projektet ska bygga på en tydlig designprocess och inkludera:
- Användarstudie (minst 5 svar, t.ex. från klassen). Se Webbriktlinjer.
- Personas (minst 1 baserad på studien).
- User stories kopplade till personas.
- Sitemap och lo-fi wireframes/prototyper.

Tekniska krav

- Applikationen ska deployas hos en leverantör som stödjer Node.js (tänk på CORS och HTTPS).
- Backend ska implementeras i Node.js med Express eller Fastify (annat val kräver godkännande).
- Databasen ska vara MongoDB.
- Frontend bör byggas i React eller Angular.
- Projektet bör göras tillgängligt offline (PWA).
- API-dokumentation bör finnas (t.ex. via Insomnia).

För högre betyg är följande meriterande:

- Testdriven utveckling (TDD).

Arbetsprocess

- Projektet måste versionshanteras i Git. Inlämningar med endast en commit kommer inte godkännas.
- Följ GitHub Flow som arbetsmetodik.
- Koden ska följa kodstandarder:
  * Airbnb JavaScript Style Guide, https://airbnb.io/projects/javascript/
  * Airbnb CSS Style Guide, https://github.com/airbnb/css
- Använd en konfigurationsguide, t.ex. ESLint + Airbnb Style + Prettier. https://vicvijayakumar.com/blog/eslint-airbnb-style-guide-prettier/

Inlämning

- Projektet lämnas in som en länk till GitHub-repot: GitHub Classroom-länk
- Läraren måste ha åtkomst till repot.

Förslag på arbetsplan

1. Iteration 1: UX/design och användaranalys.
2. Iteration 2: Bygg vyer och komponenter (frontend), koppla mot enkel mock-backend.
3. Iteration 3: Implementera backend i Node (databasstruktur + autentisering). Testa routes i Insomnia.
4. Iteration 4: Koppla frontend mot backend. Börja testa deploy.
5. Iteration 5: Testa noggrant (i olika webbläsare). Deploya skarpt.
6. Iteration 6: Fixa buggar och/eller lägg till extra funktionalitet för högre betyg.

Tips: det går också bra att börja med backend och ta frontend senare.

Förslag på projektidéer

• Webbshop
• Bokningssystem
• Fotoapplikation
• Twitter-klon
• Webbaserad emulator

Bedömning

Kursmål som examineras

- Använda JavaScript-ramverk och -bibliotek
- Utveckla avancerade webblösningar i JavaScript
- Utveckla enklare backendlösningar i JavaScript
- Utveckla Progressive Web Apps
- Arbeta som fullstack-webbutvecklare i JavaScript

Betygskriterier

Godkänd:
- Alla kursmål ska uppfyllas med gott handlag.
- Kod ska enhetligt följa angivna kodstandarder.

Väl godkänd:
- Alla kursmål ska uppfyllas med mycket gott handlag.
- Inlämningen ska innehålla utförlig och enhetlig dokumentation (både i kod och separat).
- Kodbasen ska vara väl refaktorerad och strukturerad.