

Arbeidskrav 1

7. Februar - 14. Februar

Arbeidskrav 1 i Javascript og API

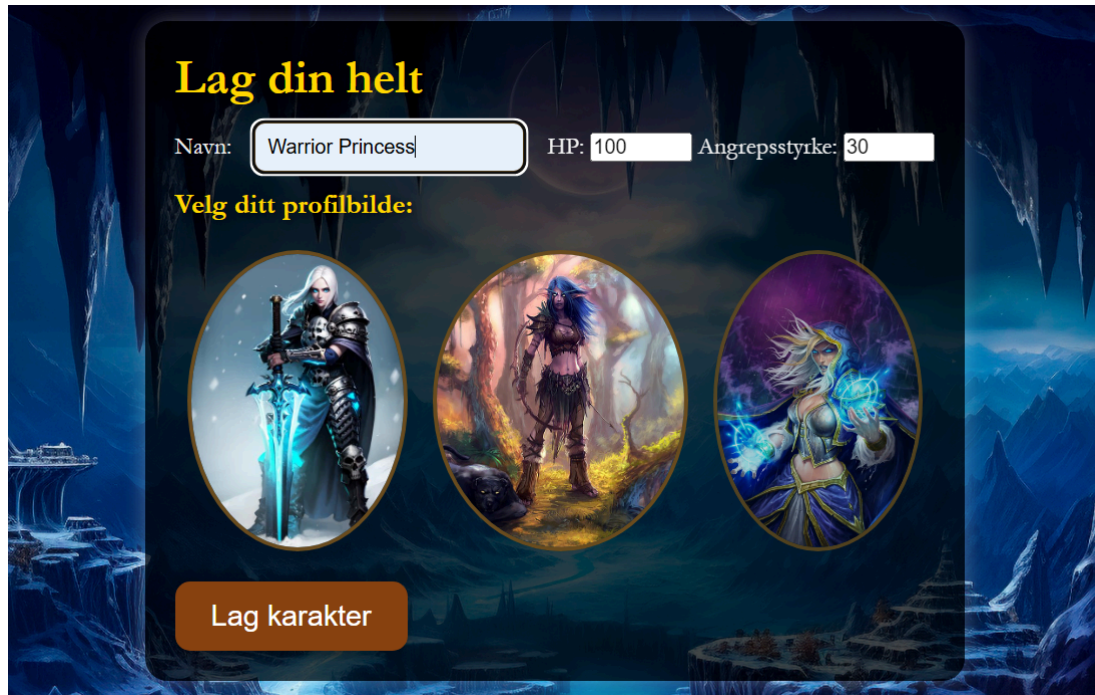
- Besvarelsen skal arbeides med og leveres individuelt
- **Alle hjelpemidler er tillatt, men merk at kopiering av kode fra AI eller nett er ikke tillatt.**
- Besvares med **zippet mappe** som inneholder: Kodebesvarelse, .git-fil, word/txt-fil med github-navnet ditt, og videobesvarelse
- Besvarelsen leveres i Wiseflow
- Datointervall er 7. Februar - 14. Februar

Kriterier til oppgaven

- Kun Vanilla Javascript er tillatt
- Det skal være et node-prosjekt
- Det skal benyttes Jest for testing. Du kan bruke jest-dom hvis du ønsker.
- Du kan benytte Parcel eller Babel, men du kan også bare bruke live-server.
- Læreren som retter oppgaven skal kun måtte kjøre "npm install", "npm test" og deretter enten kjøre babel, parcel eller live server for å se på programmet i nettleseren. Det skal ikke være nødvendig å jobbe seg gjennom feilmeldinger e.l. for å få kjørt applikasjonen, den skal fungere "out of the box" etter npm install..
- Du skal pushe og commite til github. Ha med .git-filen i den zippede mappen.
- Videobesvarelsen **skal ikke overstige 5 minutter**. Hvis videoen er **05:01 så er videoen for lang**. Du skal snakke om de elementene av prosjektet ditt som du synes er viktig å vise frem. Det er viktig at ansiktet ditt synes i videoen, og at lyden er tydelig. Videoen skal ikke speedes opp.

Oppgavetekst

I denne oppgaven skal du lage en fantasy-karakter som kan kjempe mot en tilfeldig fiende. Du får ferdig HTML-kode (se den zippede mappen) og skal kun fokusere på å implementere JavaScript-funksjonaliteten og skrive tester for applikasjonen. Du skal bruke vanilla JavaScript for å manipulere DOM, lagre data i localStorage, og skrive unit-tester ved hjelp av Jest.



Del 1: Lag karakteren din

- 1. Opprett en karakter:**
 - Karakteren skal ha et navn, HP (helsepoeng), og angrepsstyrke (attack damage).
- 2. Velg et profilbilde:**
 - Karakteren skal kunne velge et profilbilde fra 3 tilgjengelige bilder.
- 3. Lagre karakteren:**
 - Når brukeren klikker på "Lag karakter"-knappen, skal karakterens navn, HP, angrepsstyrke, og profilbilde lagres i localStorage. Hvis man trykker refresh, vil altså karakteren fortsatt ligge i localStorage.



Del 2: Generer en fiende

1. Generer en tilfeldig fiende:

- Når brukeren klikker på "Generer fiende"-knappen, skal en tilfeldig fiende bli generert med:
 - Et tilfeldig navn (Goblin, Ork, eller Drage).
 - Et tilfeldig bilde
 - HP mellom 50 og 150.
 - Angrepsstyrke mellom 10 og 40.
- Fiendens data skal lagres i localStorage.



Del 3: Sloss!

1. Simuler en kamp:

- Når brukeren klikker på "Start kamp"-knappen, skal det sammenlignes hvem som har høyest HP etter begge karakterene har gjort et angrep, og resultatet skal vises:

- Hvis spillerens HP er høyere enn fiendens, skal det vises "Du vant!".

- Hvis fienden har høyere HP, skal det vises "Du tapte!".

- Hvis HP er like, skal kampen ende uavgjort og vise "Uavgjort!".

2. Vis karakter- og fiendens profilkort:

- Du skal vise et kort med informasjon om både karakteren og fienden, inkludert navn, HP, angrepsstyrke og profilbilde. Informasjonen skal vises basert på hva som ligger i localStorage.

Del 4: Testing

Skriv tester:

- Skriv minst 3 Jest-tester for programmet ditt. Noen eksempler på funksjonalitet du kan teste er:
 - Lage karakter: Test at karakteren blir opprettet med riktige verdier.
 - Generere fiende: Test at fienden blir generert med de riktige verdiene.
 - Kampresultat: Test at kampen viser riktig resultat basert på HP.
 - Lagring i localStorage: Test at både karakteren og fienden blir lagret riktig i localStorage.