Examination: Individuellt Projekt: “FilmSamlaren”

Översikt

Som en avslutande examination i Frontendutveckling har du fått uppgift att utveckla Hemmakvälls nya applikation: "FilmSamlaren".
Den applikationen skall demonstrerar dina kunskaper och färdigheter inom:

- JSON-hantering
- HTTP/HTTPS-förfrågningar
- Asynkrona funktioner (async/await med try/catch)
- Dynamisk webbutveckling med HTML, CSS, JavaScript (inkl. datatyper, operatorer)
- Grundläggande UX/UI-principer med hänsyn till WCAG (via designskiss i Figma och viss anpassning i din kod)

Syfte

Målet är att du ska tillämpa kursens innehåll i ett sammanhängande, individuellt projekt. Genom projektet visar du att du kan:

- Hämta och hantera data i JSON-format.
- Förstå och använda HTTP/HTTPS genom en asynkron förfrågan mot ett offentligt API.
- Skriva asynkron JavaScript-kod med async/await och hantera fel med try/catch.
- Bygga en dynamisk komponent på webbsidan.
- Använda datatyper och operatorer korrekt i JavaScript.
- Demonstrera förståelse för UX/UI-principerna och WCAG genom att först ta fram en designskiss i Figma och sedan implementera lösningar i koden som främjar tillgänglighet och god användarupplevelse.

Projektbeskrivning

FilmSamlaren är en webbapplikation som låter användaren söka efter filmer genom att använda ett offentligt film-API (t.ex. The Open Movie Database (OMDb)). Användaren ska kunna:

- Se minst 10 filmer utan att behöva söka efter något.
- Ange en filmtitel i ett sökfält eller användaren skall kunna filtrera filmer med via exempel genrer.
- När användaren klickar på en vald film, visa mer detaljerad information (titel, år, genre, kort beskrivning, poster m.m?).

Du ska utveckla applikationen individuellt.

Projektkrav

Teknisk Funktionalitet för att nå G nivå:

1. **Datahantering (JSON):**
    - Hämta data i JSON-format från ett offentligt API.
    - Tolka och presentera JSON-datan på webbsidan.
2. **HTTP/HTTPS & Asynkronitet:**
    - Använd `fetch()` för att göra asynkrona GET-förfrågningar mot API:et.
    - Använd `async/await` och `try/catch` för felhantering av asynkrona operationer..
    - Säkerställ att du hanterar nätverksfel, ogiltiga sökningar och oväntade svar på ett användarvänligt sätt.
3. **Dynamisk Komponent och Datatyper/Operatorer:**
    - Låt användaren **söka** efter filmer via ett input-fält **eller** en **filtrering** av listad data.
    - Visa **sökresultatet** eller **filtrerad** data dynamiskt på sidan (skapa element i DOM).
    - Räkna och visa antalet matchande filmer. Använd JavaScript-logik (t.ex. operatorer och villkor) för att dölja visa element vid behov.
    - När användaren klickar på en film, visa detaljer (på samma sida eller i en modal/overlay).
    - När användaren interagerar (t.ex. skriver in sökord, klickar på en filter knapp) ska innehållet på sidan uppdateras dynamiskt.
4. **UX/UI och WCAG**
    - Ta fram en enkel wireframe eller mockup i Figma där du visar hur du tänkt kring layout och UX/UI-principer (de 6 principerna ni lärt er).
    - Bifoga en länk eller bild på din Figma-design i projektets README.
    - Implementera minst några grundläggande WCAG-hänsyn i din kod (ex. alt-texter på bilder, tydlig färgkontrast, semantisk HTML-struktur).
    - Använd principer om användarvänlighet, konsekvens, feedback, felprevention – t.ex. ge användaren tydlig återkoppling när data laddas, eller vid fel.
5. **Versionshantering med Git:**
    - Använd Git för att versionshantera ditt projekt.
    - Minst 3 meningsfulla commits.
    - 1 “huvudbranch” (main?), den använder ni vid presentationen.
    - 1 develop branch som mergas i huvud branchen.
    - Minst 1 feature branch som mergas i developbranchen.

Teknisk Funktionalitet för att nå VG nivå:

1. **Favoritlista (LocalStorage):**
   Låt användaren markera utvalda filmer som favoriter och spara dem i LocalStorage. Favoriter ska kunna ses även efter om laddning av sidan.
2. **Responsiv Design:**

Gör gränssnittet responsivt. (3 skärmar: mobil, tablet, desktop storlekar)

3. **UX/UI och WCAG**
   Figma design för alla 3 skärmar
4. **Hanterar asynkronitet med gott handlag:**
   Använder Promise.all() eller caching.
5. **Versionshantering med Git:**
   1 huvud branch, minst 1 develop branch och minst 3 feature brancher och allt mergas i rätt ordning (feature branch inte in direkt i huvud branch), minst 5 commits, tydliga commit-meddelanden.
6. **Skapa minst 2 sidor.**
7. **Organisation:**
   Har du flera filer av samma sort? Skapa mapp för det, till exempel du har 2 html filer, skapa mapp!

Inlämning

1. **Kod**: Lämna in GitHub-länk till läraren i Canvas (Modul Examination -> Flik: Slutbetyg: Frontendutveckling).
2. **README.md**:
   - Kortfattad beskrivning av projektet och hur man kör igång det lokalt.
   - **Länk** till din Figma-skiss.
   - Kortfattad förklaring av hur du uppfyllt JSON-, HTTP/HTTPS-, asynkronitets- och UX/UI-kraven.
   - Beskriv hur du hämtar data från API:et (Vilket API? URL/enpoint, parametrar, API nyckel?).
   - Hur man navigerar/använder applikationen.
3. **Presentation (ca 8-10 min)**:
   - Visa applikationen i webbläsaren.
   - Förklara hur du hämtar och presenterar data från API:et.
   - Påvisa var du använt async/await och try/catch.
   - Visa din Figma-skiss och förklara hur du tänkt kring UX/UI och WCAG.
   - Diskuterar hur Git använts i projektet.
   - Om du siktar på VG, visa de extra funktionerna du lagt till.
   - Var beredd på frågor.

Deadline och Presentation

- Inlämning av kod och README: Måndag 2025-01-06, klockan 23.59
- Individuell presentation: Boka er inne i detta dokument: https://docs.google.com/document/d/1AkVprz2sUEwPJ8i4wRPQP-euUTh8DIT-yDS4CmEBtPo/edit?usp= , Förnamn + efternamn, “först till kvarn”.

Komplettering?

Kompletering ifall ni inte lyckas nå G nivå finns det chans att genomföra kompletering som ska vara inlämnad inom 3 veckor. Lärare kommer kontakta den som behöver kompletera något.

Tips

- Börja enkelt: Hämta data och visa på sidan. Lägg sedan till sök/filter.
- Testa fetch-anrop och asynkron kod tidigt.
- Använd devtools för att felsöka.
- Titta på Figma-tutorials för att snabbt göra en **enkel** wireframe.

Lycka till med examinationen!