

Frontend 2 - Inlämningsuppgift

Loggbok

2026-02-02

Påbörjade arbetet med projektet. Jag tog mig först tid att börja planera och försöka se helheten gällande appen som ska byggas. Då jag kommer att bygga appen i React har jag med papper och penna skissat ner hur jag ska bryta ner appen i mindre komponenter samt för vilka komponenter state kan ligga lokalt och i vilka fall det bör hållas högre upp i kedjan. Lite tankar hur appen eventuellt kan växa genom att längre fram skapa ett eget backend api som kan sköta all business logic.

Efter detta har jag kommit igång genom att skapa ett nytt React-projekt med Vite. Jag har sedan installerat json-server och fyllt "databasen" med några filmer. Därefter har jag skapat en Movie-klass i Javascript som kan användas för att mappa datan som kommer från api:et till standardiserade objekt.

Efter detta har jag skapat en komponent som jag kallar MovieSelector som låter användaren välja vilken film de vill boka platser för. MovieSelector hämtar data om filmerna från api:et, mappar datan till Movie-objekt och fyller select-listan med denna info. När användaren väljer film uppdateras state hos parent-komponenten "App".

2026-02-04

Lagt till MovieTheater komponent som innehåller skärmen och alla platser. Platserna genereras dynamiskt baserat på prop size (rows x seats) och för varje seat skapas ett element/objekt av komponenten Seat.

Seat som är bokade markeras som occupied och resterande platser går att markera/avmarkera.

Lagt till komponenten SelectionSummary som räknar ihop antalet markerade platser samt kostnaden för dessa och visar detta för användaren.

Lagt till SeatExplanation komponent som visar för användaren hur en ledig, markerad och upptagen plats ser ut.

Lagt till BookingForm komponent som visas när användaren klickar på knappen reservera. I formuläret får användaren mata in namn och telefonnummer. När form

submittas skickas request till json-server api med användarens inmatade input samt filmens id, markerade platser och kostnad.

Utöver en post-request till booking görs även en patch-request till movie där de nu bokade platserna läggs till i den valda filmens bookedSeats. Detta ska i vanliga fall skötas av backend, men då vi i detta fallet använder json-server fick denna logik skötas av frontend.

2026-02-05

Kort dag.

Jag har slutfört arbetet på BookingForm och mergeat in branchen i main. Detta inkluderar bland annat tillägg av funktion för att validera user input med javascript samt hantering och rendering av felmeddelanden vid felaktig input eller bekräftelse av lyckad bokning.

2026-02-10

Kort dag.

Jag har påbörjat arbetet med admin-sidan. Innan jag började med det valde jag dock att göra en liten refactoring där jag bröt ut logik för hämtning av filmdata från api till egen funktion och fil. Detta håller koden mer DRY när jag kommer att använda samma logik även på admin-sidan.

Utöver detta har jag skapat en komponent för admin sidan samt en "MovieListItem" komponent och visar info om filmen med två knappar, edit och delete. Funktionalitet för hämtning av alla filmer (och rendering av en MovieListItem-komponent för varje film) samt radering av film är implementerad.

2026-02-11

Jag har under eftermiddagen gjort klart admin-sidan. Jag har lagt till funktionalitet för att lägga till samt uppdatera filmer. Dessa två använder samma form som laddas med befintlig films data vid uppdatering eller startar tom vid skapandet av ny film. Vid uppdatering eller radering av film triggas nu även en ny hämtning av datan från api:et för att säkerställa att uppdaterad data visas.

Jag har installerat react-router-dom och satt upp router och navigering för att gå mellan admin page och huvuddelen av appen.

Utöver detta har jag även gjort lite mer refactoring av services där jag brytit ut generell fetch-logic till [api.js](#) och hanterar specifik movie och booking logik i respektive service.

2026-02-12

Dagen har för det mesta bestått av att göra klart lite smågrejer här och där innan deployment. Detta involverar bland annat lite refactoring, refresh av säten efter bokning (för att de nyligen bokade säten nu ska vara otillgängliga), lite css-finslipning, bekräftelse innan radering av film och mer. Se git log för mer. När allt detta var färdigt deployades sidan med Github Actions. Därefter uppdaterade jag readme med kort beskrivning om projektet samt steg för steg instruktioner om hur appen kan köras lokalt.

Motivering av ramverk

Jag har valt att bygga appen i React. Jag har tidigare erfarenhet av React och har byggt en del projekt med det, till exempel frontend-delen av en full-stack social media app. Detta ledde till att det kändes som ett naturligt val att köra på React för denna applikation. Samtidigt skulle man kunna argumentera att det kanske skulle kunna vara gynnsamt att testa tex Angular då jag redan har erfarenhet av React, och det hade absolut kunnat vara ett alternativ. Anledningen till att jag valde att inte göra det är just Reacts popularitet. React är idag det mest populära bland alla Javascript ramverk/bibliotek och jag motiverade valet med att det därmed aldrig skadar att få lite mer övning och timmar bakom sig.

Jag valde att skriva i vanlig Javascript och inte Typescript. Anledningen är rätt simpel, jag hade ingen erfarenhet eller särskilt mycket koll på Typescript när jag bestämde mig för tech-stacken. Dock har jag nu hunnit kika på Typescript och ångrar lite att jag inte valde att implementera det från början då jag nu funderar på att migrera projektet till Typescript, efter inlämning.