

Post Mortem / Slutrapport

Dressup

Beata Eriksson 2023-06-02

Abstract

Dressup är en garderobstjänst där du som användare kan ladda upp bilder på dina kläder och sedan bygga och spara dina egna klädkombinationer. Denna rapport går igenom och sammanfattar utvecklingsprocessen och projektet i stort. Dressup är en driftsatt webbapplikation, där utvecklingen av den har kantats med en del problem och många lärdomar.

Inledning och bakgrund

Syfte / mål

Målet med projektet var att utveckla en webbapplikation där man som användare laddar upp bilder på sina egna klädesplagg och sorterar in dem i kategorier. Användaren kan se sina sparade plagg och visa plaggen inom respektive kategori. Målet var att man som användare ska kunna bläddra bland sina plagg för att hitta och sätta ihop klädkombinationer. Det finns applikationer idag som erbjuder funktionalitet för att ladda upp och spara sina kläder, vad jag upplevde saknades (och som jag ville komplettera med detta projekt) var en plattform där man kan bläddra bland plaggen - med en tanke om att det skulle påminna om hur man bygger, eller väljer kläder till, en spelkaraktär.

Webbapplikationen har en outfit-builder funktionalitet, där man kan bläddra mellan plaggen i varje kategori för att slutligen hitta en klädkombination som man tycker om för att då spara den och kunna se sina sparade klädkombinationer.

Mål kring själva lärandet eller utvecklandet av mig själv som utvecklare och student i detta projekt har varit att lära mig att utveckla i ramverket [React](#) samt hur jag kan arbeta med CSS-verktyg för att stila mina React-komponenter.

Arbetssätt

Arbetet har varit upplagt veckovis enligt hur vi inom kursen har presenterats för ett SCRUM-arbetssätt där arbetet planeras och utvärderas i iterationer, i detta fall har varje iteration inneburit en vecka. Kraven för projektet har sammanställts och prioriterats och kopplats samman med tester. Arbetet har inneburit en iterativ planeringsprocess där jag inför varje iteration har planerat arbetet, vilket omfattat specifika uppgifter med tillhörande tidsuppskattning. Uppgifterna uppdateras sedan med den faktiska tiden det tog, och planeringen kom att kompletteras om annat utöver det planerade kom att göras.

Genomförande / teknik

Vad som först och främst blev akutellt för genomförandet av projektet var att lära mig ramverket React. För att göra det har jag dels tagit del av den utbildning / information som vi har fått inom kursen, sedan har jag tagit del av tutorials och främst lärt mig ifrån <https://www.youtube.com/watch?v=98BzS5Oz5E4> (<https://github.com/iamshaunjp/MERN-Stack-Tutorial>) för en grundläggande genomgång av hur ett React projekt, som till grunden kunde påminna om mitt, faktiskt kan bli till.

Applikationen grundar sig på funktionalitet för att kunna skapa ett konto, för att enbart du som användare ska ha tillgång till dina uppladdade bilder. Webbapplikationen är en PWA (Progressive Web Application), vilket innebär att den går att ladda ner som en applikation till sitt skrivbord eller sin dock. Jag har valt att utveckla mitt projekt med två olika kodbasar. Jag kommer nedan att gå igenom de tekniker jag har använt för min frontend (klientsidan) och min backend (serversidan).

Frontend

Min frontend är implementerad i [React](#). För att stila mina komponenter har jag använt mig av CSS-verktyget [emotion](#). Jag har driftsatt min frontend kod via [netlify](#), där den med sin url blir nåbar för användaren.

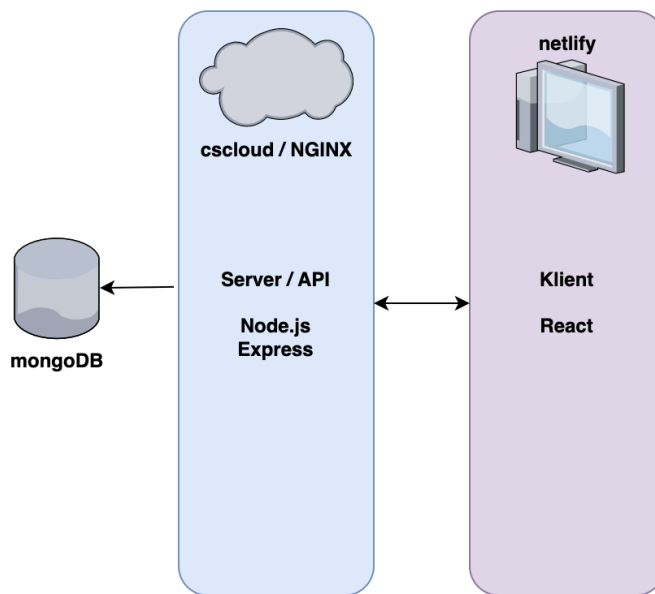
Backend

Min backend står på egna ben från min frontend och kontakten dem emellan sker genom anrop från min frontend. Min backend är implementerad som ett RESTful API skriven i [Express](#) och [Node.js](#). Mitt API besvarar klientsidans anrop över HTTP med data i JSON-format och erbjuder olika endpoints, URLer, och metoder för att nå olika resurser.

Min applikation har funktionalitet för att skapa ett konto och jag har valt att göra det med JWT (Json Web Token). För att användaren ska kunna ladda om sidan i webbläsaren utan att behöva logga in på nytt så sparas denna token som en cookie i webbläsaren. Denna cookie hanteras, skapas och raderas från serversidan.

Arkitektur

Min backend är driftsatt på cscloud/NGINX och det är genom detta som min klientsida kan göra anrop mot mitt API. Servern är konfigurerad mot [mongoDB](#), den databas som hanterar all lagring data (användarens uppgifter och bilder) som applikationen kräver. Nedan är en övergripande illustration för projektets arkitektur.



Positiva erfarenheter

Att arbeta med React och komponenter har varit väldigt utvecklande. Jag har upplevt det som ett behagligt och tydligt sätt att skriva och separera koden på. Jag har tidigare inte skrivit min backend som ett API eller haft olika projekt för min frontend respektive backend. Detta har gjort det tydligare för mig hur olika delar av ett system kan samverka och separeras och vilken del av vilken kod som gör vad.

Det var givande att lägga upp en tydlig planering för varje iteration för att få tydlig överblick över vart man låg i processen och vad man slutfört. Att ha veckovisa möten med sin tilldelade grupp har också fått en att vilja prestera och se till att ha någonting att presentera för gruppen. Det har också varit lärorikt att träna på att kunna sammanfatta sin vision och vart man är i processen.

Det har också blivit klarare för mig hur man kan hantera autentisering med JWT, vilket jag antar att jag kommer ha nytta av efter detta projekt. Min upplevelse av utvecklingen i min frontend-projekt har generellt varit positiv, jag har varit nöjd med att arbeta med React, Emotion och Netlify och skulle gärna se till att göra det igen.

Negativa erfarenheter

Trots att min huvudsakliga upplevelse av projektet varit positiv har jag som förväntat ibland hamnat i vad som känts som loopar utan slut. Exempelvis att skapa serverrenderade kakor visade sig vara en tidsslukare som jag inte hade räknat med. Men även här blir det klart för en när problemet löst sig, att man tar med sig lärdomar ifrån det.

Jag har också behövt ta avsteg från min befintliga vision där tanken var att använda mig av ett externt API där varje uppladdad bild blir av med sin bakgrund. Men de APIer jag hittade som erbjuder denna tjänst krävde alla någon typ av betalning. Jag ansåg inte detta vara fatalt för applikationen och tog beslutet att arbeta vidare med projektet och slutföra min vision utan detta steg. Jag kommer ta med mig att se till att utforska externa resurser innan visionen skapas, för att redan då veta vad jag kan och vill arbeta med - eller inte.

Det har stundvis känts tungt att dokumentera sin testning, sina krav och sin planering. Jag kom att se det som en bra övning för framtida projekt då jag är medveten om att denna typ av dokumentation krävs för att projekt ska kunna överleva när man arbetar tillsammans med andra.

Jag har behövt komplettera min dokumentation under projektets gång. Dels har jag insett att en del krav ursprungligen inte var helt mätbara samt att testerna inte alltid har täckt det krav som det avsågskligen skulle testa. Detta löste jag genom att omvärdera och omformulera mina krav vid behov och skapa tester utefter att jag kom på hur jag bättre bör och kan testa. I efterhand ser jag hur jag hade kunnat göra annorlunda redan från början, att i ett tidigt skede av projektet skapa testfall för varje krav och redan då hitta eventuella brister i kravspecifikationen.

Sammanfattning

Mitt projekt i form av den slutliga applikationen som blev till fungerar. Det är inte en webbapplikation som är redo att ta över världen men det är en webbapplikation som jag har skapat i denna kursen. Jag hade gärna utvecklat produkten ytterligare, skapat bättre bildhantering samt funktionalitet för användaren att ha en avatar. Projektet har känts levande och det har varit en ständig process, som vi blivit tillsagda under projektets gång - att om någonting inte fungerar, gör annorlunda.