

Projektspecifikation

Joachim Mattsson Lummer – 12 juni 2020

Tillämpad Programmering DVGB06, ST20

Beskrivning

Projektet är tänkt att bygga på en idé jag har haft länge. Under tiden som jag studerar har jag antecknat den effektiva tiden jag studerat: hur lång tid jag spenderat med vardera kurs per dag och hur många pluggtimmar jag har i veckan. Detta har skett för hand eller bara i fritext.

Jag vill bygga ett gränssnitt som tillåter mig att enkelt fylla i datat och som sparar det i en databas. I gränssnittet skall jag kunna sortera per kurs/vecka/månad. Total tidsåtgång ska tydliggöras på ett enkelt sätt. Datainmatningen skall vara enkel även den.

Tidigare erfarenheter och teknik

Jag är snart examinerad interaktionsdesigner med frontend nära till hands. Denna vår har jag börjat lära mig React.js och tycker det är väldigt roligt att koda i. Det tillsammans med CSS för stilsättning kommer vara en stor del av denna kurs tänker jag.

Jag har petat på Node/Express.js förut och tänker att det kommer tillgängliggöra kopplingen till databasen, vilket troligtvis kommer att skrivas i MongoDB. Det finns många bra källor som lär ut **MERN**-stacken, d.v.s. **M**ongoDB, **E**xpress, **R**ead, **N**ode. Därför lutar jag åt det hållet. Detta kommer sedan manifesteras som en webbplats, kanske att jag lägger upp den på Netlify eller liknande site.

Tidsplan

Detta är jättesvårt och kommer aldrig stämma hur mycket jag än funderar över det. Jag satsar på konsekvent 30 timmar i veckan, där jag också ger mig själv utrymme till reflektion som ju

sker oavsiktligt utanför dessa timmar. Eftersom jag inte kodat en fullstack webbapplikation så är det svårt att estimerar tidsåtgång för Backend och Databas.

| Område | V1 – Prototyp | Tid hr |
|---------------------|---|--------|
| Gränssnitt | Grundlig funktionalitet där inmatningskomponent lagrar datat i ett state och tabellkomponenten visar datat från state | 30 |
| Stilsättning | Grundlig layout för inmatnings- och tabellkomponent, främst marginaler och storlekar | 5 |
| Backend | Express ska skicka/hämta arbiträr data genom API | 20 |
| Databas | Se över möjligheten till att spara data i MongoDB. Bestäm datatyper och schema(?). | 20 |

| Område | V2 – Körbar version | Tid |
|---------------------|--|-----|
| Gränssnitt | Inmatningskomponenten skickar data till Express API och är utformad på ett smidigt sätt för enkel inmatning. Tabellkomponent hämtar data från Express API. | 40 |
| Stilsättning | - | - |
| Backend | Express är kopplad till mongoDB där det genom ett API går att spara och hämta data. API:et ska gå att nå via gränssnittet. | 40 |
| Databas | Det går nu att hämta och spara data på ett tillförlitligt sätt. | 30 |

| Område | V3 – Färdig produkt | Tid |
|---------------------|--|-----|
| Gränssnitt | Tabellkomponent kan filtrera och sortera data. Sömlös interaktion. Inloggningssida. | 20 |
| Stilsättning | Snygg responsiv design som överensstämmer med kvaliteten på en slutgiltig produkt. Tillgänglig och användbar. Figur som är procentuellt ifylld beroende på markerad ihopslagen data. | 40 |
| Backend | Inloggningsfunktionalitet. | 30 |
| Databas | Inlogging vilket möjliggör användarspecifik datahämtning. | 30 |

Det finns ambitioner som jag kommer få brottas lite grann med. Exempelvis:

- Hur datat ska tillgängliggöras för specifik användare, d.v.s. någon login funktion?
 - detta är egentligen helt utanför mitt område, frontend och interaktionsdesign.
- Spinna vidare på den vedertagna idén om att det behövs 10,000 timmar för att bli *bra* på någonting, där en idé jag har är ett symboliskt träd som växer ju fler timmar som registrerats.
 - INTE en key feature, troligtvis utanför kursens omfång.

Skiss

