***The Proof Of Concept rapport av Jonas Rosberg*** *Githup Repo:* <https://github.com/joro1993/Create-A-ScoriaAB/tree/SecondVersion>Min inlämning, ScoriaAB är i samma repo som inlämning 1 – en god start. Det finns två branches. Master branch innehåller version 1, alltså från inlämning 1. Version 2 (SecondVersion som branchen heter) är den nya versionen. Version 1 är tillgänglig i SecondVersion också. Men App.html visar numera en Single Page Applikation som man kommer åt via en knapp på index.html, index.html är sidan som startas när man först startar upp programmet med npm start.

Det finns en README.md fil tillsammans med källkoden som ger en tydlig beskrivning hur man startar upp projektet så att den visas på webbläsaren.  
  
Projektet i SecondVersion-branchen innehåller 10 komponent-filer om man räknar med App.js och orders.js.  
Mitt projekt blev inte riktigt som jag hade tänkt mig, fick ta bort en del grejer så att projekten kunde köras. Jag är inte nöjd med inlämningen då jag hade högre förväntningar. Men i brist på tid kvar innan inlämnningsdatum så fick det bli som det blev nu.   
Så orders.js komponentet är helt bort-kommenterad, den är inte aktuell längre. Och räknar man bort App.js så har jag 8 komponenter.  
Dessa är: business.js, Contact.js, home.js, menu.js, myIdea.js, products.js, prototype.js och worker.js.   
Dem flesta ingår i min Router jag har i menu.js komponenten för att skriva ut mina sidor.   
  
Jag installerade JavaScript biblioteket Redux. Redux installerades via bash terminalen med kommandot *npm install redux react-redux.*   
Redux användes för att kunna manipulera data/information i projektet, och i detta fall är den datan/informationen mina produkter/tjänster som finns på hemsidan.  
  
***d1 - Förklara med egna ord vad som är viktigt att tänka på vid skapandet av webbapplikationer med React (ungefär tre paragrafer).***  
React är ett frontend JavaScript bibliotek. Om man vill använda något för backend till React så är NodeJS ett bra alternativ, men för just React är det frontend som gäller!  
React använder även komponenter, så man smidigt kan göra single page applikationer.   
  
***d2 - Beskriv och förklara användningsområdet av det ramverk/bibliotek utöver React som du använt i din applikation.***Redux användes i mitt projekt för att manipulera data, jag använder React för att kunna få ut mina produkter och tjänster. Jag använder Provider som hämtas från react-redux. Jag skapar även en const variabel som heter store för att skapa en createStore som hämtas från react. För att kunna passera min data och skapa en ”store”.  
  
***d3 - Förklara med egna ord vad som är viktigt att tänka på vid användningen av lokal data i React (ungefär en paragraf).***Att hämta data från en lokaldata källa för att kunna återkoppla information och data till React’s tillståndshantering är det viktigaste man bör ha koll på vid användning av lokal data i React. Ett RESTful Api är ett exempel på en extern datakälla man kan använda, som man får hämta ut med hjälp av Fetch.

***r1 - Vilka åtgärder har du genomfört i din källkodshantering för att kunna rulla tillbaka och vidareutveckla projektet i framtiden? Varför är dessa lämpliga val?***För att kunna rulla tillbaka till första versionen så finns det två olika brancher. Master branch innehåller original-versionen. SecondVersion innehåller andra versionen, som är skapat och skrivet i React och Redux. Original-versionen innehåller varken React eller Redux.  
  
Anledningen varför detta är en lämplig metod är för att det är väldigt smidigt att bläddra fram och tillbaka.   
Om du är inne i masterbranch så ser du det längst ner i vänstra hörnet, för att bläddra till en annan branch så klickar du på masterbranch och väljer SecondVersion-branchen för nya versionen. Om du är inne i SecondVersion så ser du det på samma sätt, längst ner i vänstra hörnet. Klicka på SecondVersion och välj masterbranch för att gå tillbaka.  
  
***r2 - Gällande det bibliotek/ramverk utöver React som du lagt till i ditt projekt - vad tycker du detta paket tillför för styrkor till ditt projekt och vilka begränsningar introducerar det för framtida designval?***Av den lilla erfarenhet jag nu fått av Redux, så verkar det vara mycket smidigt att manipulera data.   
Jag tror att om man förlitar sig för mycket på JavaScript bibliotek som Redux för att hjälpa en, så begränsar man sig själv lite. Det kan vara bra att tänka på att lära sig grunden först, innan man tar sig åt hjälpmedel.   
Precis som med CSS och Boostrap. Jag tror det är viktigaste att kunna grunden först, alltså CSS.  
  
***r3 - Analysera och reflektera kring den eventuella byggprocessen du använt för att kompilera JSX koden i ditt projekt - vad var bra och dåligt med att välja denna metod?***Jag använde exempelkoden från Viktors repo som han tipsade mig om, det tog ett litet tag innan jag fick React att funka och kompilera som det ska. Men jag upplevde att det blev lite krångligare än vad det behövde vara. Jag har testat att skapa många olika react-projekt på min fritid som hjälpte mig, jag använde create-react-app till dessa.  
  
I mitt projekt, så var det en del kod som funkar utmärkt i mina create-react-app applikationer, men som i mitt projekt visar errors. Jag hade ingen aning om hur jag skulle göra här.   
Sen så visar mitt projekt en del varningar i console.   
Hemsidan funkar och körs dock.   
Men funderingar kring om jag ska köra en create-react-app och flytta över mitt projekt istället har uppstått några gånger.   
   
Det var en bra erfarenhet att försöka lägga till React i mitt redan existerande projekt, och det är något jag planerar att kolla upp lite djupare.  
  
***Framtiden för mitt projekt***Planen var att skapa en varukorg för att kunna lägga in mina tjänster och göra beställningar. Detta funkade inte alls. Tyvärr. Så jag ska försöka fortsätta implementera denna varukorg som jag gärna vill ha. Sen stilmässigt, så ser hemsidan inte bra. Jag har lagt till lite grundläggande inline styling. Men jag kommer nog i framtiden lägga till Bootstrap eller Materialize för stil.