**Kunskapskontroll 2 - Invoice-dashboard**

Förfaller 25 november 2022 kl. 23:59

**Instruktioner**

I Kunskapskontroll 1 byggde ni en tidtagnings-applikation, där man kunde lägga in projekt & tasks och ta tiden när man jobbar på en task.

Denna gången kommer ni bygga andra delen av den här applikationen - en "Fakturerings-dashboard".

Och kommer byggas som en traditionell Admin-dashboard, med responsivt med stöd för **både mobil och desktop.**

Frontenden ska skrivas med **React och Typescript. React Context** ska även här användas för att lagra data som används på flera ställen i applikationen.

Vi kommer fortsatt använda **json-servern**och datan vi skapar med Kunskapskontroll 1.  
Ni väljer om ni vill bryta ut den eller köra den direkt från K1-mappen. Detta för att fortsatt kunna logga tid med K1 och visa den datan i vår Dashboard.

Exempel på datastruktur: <https://gist.github.com/sawariz0r/536ccefb4729d079277d7062a632ddfb>

Exempel och inspiration på design för en Dashboard kan ni bland annat hitta på: <https://dribbble.com/search/dashboard>

En användare ska kunna:

* Se en lista på tidtagningar de senaste 30 dagarna  
  (Har du byggt in dessa i tasks: tasks som loggat tid de senaste 30 dagarna)
* Se en lista på alla tidtagningar
* Se en lista på projekt
* Se en lista på tasks
* Ta bort en tidtagning
* Ta bort en task
* Ta bort ett projekt
* Sätta ett timpris på ett projekt
* Skapa en "faktura" för ett valt projekt och välja tasks som "fakturan" ska innehålla
* Se en lista på "fakturor" och status
* Se en överblicks-sida som visar:  
  - Antal projekt  
  - Antal tasks  
  - Antal "fakturor"  
  - Tid som loggats de senaste 30 dagarna  
  - Antalet kronor som fakturerats det senaste året

För att uppnå **Godkänt**är kraven att:

* Byggd med Typescript, React som Frontend och json-server.
* Applikationen ska använda React Context som "Store" för applikations-bred data.
* Samtliga krav under "En användare ska kunna" är uppfyllda.
* Applikationen ska vara byggd med responsiv design i åtanke.
* En faktura ska innehålla  
  - Status (Ej betald, betald, försenad)  
  - Förfallodag (30 dagar från datumet vi skapade den)  
  - Summa (tid x projektets timpris)  
  - Kundens namn
* Listorna ska vara presenterade som tabeller med följande kolumner:  
  - Tasks: Namn, Projektets namn  
  - Projekt: Namn, antal tasks  
  - Fakturor: Kundnamn, status, förfallodatum, summa
* Applikationen ska inte innehålla några "@ts-ignore", typfel och inga **implicita any**.

För att uppnå **Väl Godkänt**är kraven att 2 av följande 4 är uppfyllda:

* Applikationen ska inte heller innehålla några **explicita any**.
* Kunna välja att avrunda tiden uppåt på tasks när man skapar en faktura till närmsta [1 min, 5 min, 15 min, 30 min, 1h].  
  (Avrundningen ska ske på varje individuell task, inte på hela fakturan)
* Lägg till stöd för drag n drop för att ändra ordningen på överblicks-infon på överblicks-sidan. npm install
* npm run dev npm install
* npm run dev  
  Samt att kunna visa/dölja info-delar på överblicks-sidan.
* Skapa grafer på överblicks-infon för att visa:  
  - En "Line chart" som visar totala mängden tid som loggats de senaste 7 dagarna  
  - Valfri chart som visar beloppet som fakturerats i år (per månad)

Tips:

* Innan ni sätter igång, se över datan ni kommer använda och skriv typer för dessa i förhand.  
  (Task, Projekt, Faktura, etc)

Inlämning sker via en länk till ett **privat**Github-repository (bjud in "cme-osuka" på Github)

**Mitt arbete**

Bifoga

Nytt