# Frontend test

## Instruktioner på svenska 🇸🇪

Utveckla en enkel webbapplikation med följande tre vyer och funktioner. Du väljer själv ramverk och teknik efter vad du bedömer lämpligt, om vi inte begärt att du ska använda ett visst specifikt ramverk. Fokus bör ligga på att välja ett modernt ramverk/teknik, och att visa att du behärskar dessa och strukturerar din kod med bra arkitektur som går lätt att följa.

Programmet skall hålla all data i någon form av databas (av valfri sort, exempelvis duger local storage, fil på hårddisk, konventionell databas eller molntjänst). Denna lagring ska göra det möjligt att starta om programmet / ladda om sidan, utan att information går förlorad.

För att vi ska kunna köra din kod bör den, och eventuella beroenden, kunna installeras och startas med npm eller motsvarande utan att kräva att annan mjukvara är globalt installerad. Vi föredrar att koden skickas till oss i form av en länk till din kod på Github eller motsvarande före att få den som en .zip-fil eller liknande.

Webbappliktionen ska ha följande vyer:

**Vy 1:**

Här kan användaren skapa en person. En person har ett namn samt kan vara kopplad till ett företag (företag skapas i vy2).

**Vy 2:**

Här kan användaren skapa ett företag. Ett företag skall endast ha "namn" som attribut. Här skall användaren också kunna välja ett företag och då få alla anställda listade. I den listningen ska det även gå att ta bort en anställd från företaget. Då ska den anställde inte längre vara associerad med företaget, men personen ska finnas kvar i systemet utan en anknytning till något företag (som om man skapade en person i vy1 utan att välja ett företag).

**Vy 3:**

Här finns en lista på alla personer som inte är kopplade till företag, och ha möjlighet att koppla dem till ett av företagen som skapats i vy2.

## Instructions English 🇬🇧

Develop a simple web application / webpage with at least the three views described below. Choose what framework and/or techniques that you find suitable for the task, unless we specifically ask you to use a certain framework. The focus should be to choose a reasonably modern framework/technique and show that you know this well and structure your code with an architecture that is easy to follow.

The web app shall keep all data in some kind of database (you can choose a suitable type, and everything from local storage to conventional database or web service will do). This storage shall make it possible to restart/reload your web app/page without losing information.

For us to run a test your code it, along with any dependencies, shall be installed and started through npm or similar without requiring other globally installed software. We prefer you to send the code to us using a link to your code on GitHub or equivalent, rather than having it sent to us in a .zip-file or similar.

**View 1:**

Here the user can create a person. A person has a name and can be associated with a company (companies are created in view 2).

**View 2:**

Here the user can create a company. A company should only have "names" as attributes. Here the user should also be able to choose a company and then get all employees listed. In that listing, it should also be possible to remove an employee from the company. Then, the employee should no longer be associated with the company, but the person should remain in the system without any connection to any company (as if you created a person in view 1 without choosing a company).

**View 3:**

Here is a list of all people who are not linked to companies and have the opportunity to link them to one of the companies created in view 2.