Backend Assessment - Work Orders

The goal is to build a small RESTful API that performs CRUD operations. Feel free to use any language or framework you like. This assessment will be evaluated based on the following criteria:

- Completion: Did you complete all the requirements? This is the most important criterion.
- Fundamentals: How well is the data structured and how efficient is the code?
- Code Organization and Quality: How organized is your solution?
- Testing: Did you write some tests for the solution? How is the coverage?
- Communication: We are looking for candidates with strong communication skills. This will be evaluated based on your Readme.md file.

App

You are building the back-end of an app that has work orders and workers. A work order is a job to be completed by one or more workers. Structure your application as you would for any production app - include applicable classes, routes, persistent storage, etc.

Data Model

You can use any type of database you would like to store the data for this app. The following basic information should be stored:

```
Worker
   ● Name (String)
   ● Company Name (String)
   ● Email (String)
```

```
Work Order
   ● Title (String)
   ● Description (String)
   ● Deadline (date)
```

One or more workers are ‘assigned’ to an order. A max of 5 workers can work on one order.

API

The application needs to have the following routes:

- Create a worker
- Delete a worker
- Create a work order
- Assigning a worker to an order
- Fetch all work orders:
    - For a specific worker
    - Sorted by deadline

Deployment

Since testing your application locally in different environments may not be straight-forward, we would also like you to deploy it. You can choose how you want to do this, but include a **link of the deployed web page in your Readme.** For a free option, you can use Heroku.

If you are not able to deploy your application, explain the reason in your Readme and be sure to include detailed instructions for us to run it locally (we use MacOS).

Readme.md

Along with your submission, write a Readme.md file that:

- How to run the application:
    - The command we should use installing your dependencies, e.g: *npm install, pip install -r requirements.txt*. You can assume we already have your programming language set up in our environment, as well as any common package managers (npm, yarn, bundle, pip, etc)
    - The command we should use for actually running the application, e.g: *npm start, java Main, python manage.py serve*
- A brief (three sentences) high-level description of your project, written for a non-technical person. This will help us assess how well you can communicate.

Submission Details

Please submit your code in a compressed folder (.zip, .sitx, .7z, .rar, and .gz) on the Hatchways platform. The max submission size is 5MB.

Do not submit any built folders (like node_modules), since the compressed folder will be too large

If your submission is too big (although this should not be the case if you follow the steps above), and you can't figure out how to compress, you are welcome to email your solution to hello@hatchways.io.

Please include your name, and use the email you signed up with the Hatchways platform. Use the subject line “Backend Assessment Submission”.

Public Repositories

Please avoid posting your solution to a public repository. We understand that you may want to share projects you have worked on, but many hours go into developing our tools so we can provide a fair skills evaluation.

If you would like to keep a similar version of the assessment on a public repository to showcase your skills, then please do the following:

- Remove all references to Hatchways in the assessment
- Do not use any Hatchways APIs (replace them with an API of your own)