Implementation exercise

1. Requirements / Intro

You need to implement a simple web application that allows you to create to-do items and folders to group them. The development is divided in two phases:
- Phase 1: to-do item creation
- Phase 2: folder creation to group the to-do items

IMPORTANT CONSIDERATIONS:
- **Phase 1 is mandatory** to pass this exercise, while Phase 2 will provide extra points if done.
- **Content should be persisted in a relational database** - in-memory storage or mocks are not allowed.

2. Deliverables

To pass this exercise, in addition to the implementation, you must
- Upload a code to a public **GitHub repository** and use git properly
- Include a bash / zsh script allowing to run the app - ideally, the app should start from in a Linux/macOS environment just **running one command**. This command should setup everything that is required to run the app like, for instance, setting up a DB schema, pre-creating any config file, etc.
- Include a **README.md** file describing all the runtimes, engines, tools etc. required to run the app, with their concrete versions - for instance, MySQL 5.7, Spring Boot 2.1, React 16.10, etc.

3. Technologies

There is no restriction about the technology to be used, if the **implementation satisfies the following**:
- Structure the app as a **Single Page web Application**, i.e. frontend and backend are different apps. That is the general case when you use React, Angular, Vue.js or any other similar UI framework.
- The backend app exposes a **REST API** that is the way in which the frontend app communicates with the backend
- The backend app is **separated into layers** (e.g., Controllers, Services, DAOs/Repositories, etc.). Some of the frameworks that allow this (in addition to dependency injection) are Spring Boot (Java), Nest.js (NodeJS), ASP.NET Core (.NET), Django Rest Framework (Python), etc.
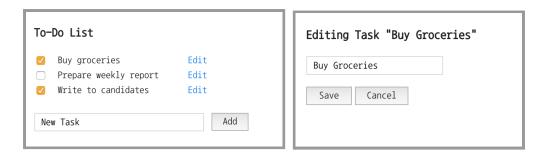
4. User Stories and mockups

Phase 1

User Stories
- As a user, I want to be able to create, edit and delete to-do items

- As a user, I want to mark/unmark to-do items as completed

Mockups

```
To-Do List

  ☑  Buy groceries              Edit
  ☐  Prepare weekly report      Edit
  ☑  Write to candidates        Edit

  New Task                    Add
```

```
Editing Task "Buy Groceries"

  Buy Groceries

  Save    Cancel
```

Phase 2

User stories
- As a user, I want to be able to create and remove new folders. NOTE: removing new folders will remove all the tasks belonging to it
- As a user, I want to be able to navigate to the item list inside a folder and manipulate the items using the same UI implemented in Phase 1

Mockups

```
Folders

- Work          View items    Remove
- Daily Tasks   View items    Remove

New Folder                Add
```

```
Folders > Work

  ☑  Prepare weekly report      Edit
  ☑  Write to candidates        Edit

  New Task                    Add
```

5. Extra functional and non-functional requirements

These features will provide **EXTRA POINTS** to your exercise results if done
- **Login**: If you provide a login screen that allows users to log in and out from the app - you need to document the default user/password used in README.md.
- **Responsive design**: If your app can be easily used on a mobile device.
- **Live deployed version**: If you deploy the app (for instance via Heroku) and you share the URL of the live running version.