Drupal technical test

Initialisation

Source code archives for D7 and D8 are provided with these instructions. They each contain a Docker configuration file that allows you to quickly set-up a development environment and a Drupal 8 website.
You are free not to use this Docker configuration and to use your local dev environment if you prefer.

What you have to deliver at the end of the test is:

- The updated source code
- An export of the Drupal database used for your tests

Drupal 7

Exercice 1

Given the following data model:

Band

- Name
- Picture
- Year of creation
- Style
- Members
- Official website

Member

- Name (firstname + lastname)
- Date of birth
- Joining date

Create the Drupal entities corresponding to this model and create a page displaying a band and his members.

The template files must be Drupal-agnostic, i.e no code specific to Drupal must be used in templates. Imagine having to implement the back-end logic yourself and then asking a front-end specialist that have no knowledge of Drupal to do the theming.
Variables used in templates must thus be processed before-hand so the template is only displaying simple variables coming from an array.

Don't focus on front-end code, the design of the final page is out of the scope of this test.

Exercice 2

Explain in broad terms how you would tackle the migration of a Drupal 7 site to Drupal 8. Assume the existing code is not state-of-the-art Drupal code.

Drupal 8

Exercice 1

Given the following data model:

Band

- Name
- Picture
- Year of creation
- Style
- Members
- Official website

Member

- Name (firstname + lastname)
- Date of birth
- Joining date

Create the Drupal entities corresponding to this model and expose the Band nodes as a REST endpoint returning the following JSON output:

```
{
 "type": "<entity bundle>",
 "name": "<band name>",
 "picture": "<picture URI>",
 "creation_date": "<date (format: YYYY-MM-DD)>",
 "style": "<music style>",
 "website": "website uri",
 "members": [
   {
     "type": "<entity bundle if any>",
     "name": "<member name>",
     "date_of_birth": "<date (format: YYYY-MM-DD)>",
     "joining_date": "<date (format: YYYY-MM-DD)>"
   }
 ]
}
```

Exercice 2

We want to be able to import some test users in our Drupal website.

Given the following csv:

```
ID,name,email,pass,roles
1,Robert Robichet,rob@appstud.com,pass4rob,authenticated
2,Marcel Patoulachi,pat@appstud.com,pass4pat,authenticated
3,Jean-Pierre Vidol,jp@appstud.com,pass4jp,authenticated
```

Import these users in Drupal using the Migrate module.

Notes:

- The passwords must be encrypted during the process so the users can log-in after import
- The import will be run using the drush migrate:import <migration name> command