D1 - Yowl

D-YOW-100

YOWL

Bootstrap

1.3

This bootstrap allows you to self-assess and estimate your strengths and weaknesses relative to the project requirements.

The points of this Bootstrap cover the main themes of the subject and are independent; they are designed so that you can assess your level in order to strengthen your knowledge in certain areas.

You don't have to do this Bootstrap all at once!
It covers several points of the topic that you will work on later.
Come back to this Bootstrap regularly throughout your project.

If I were you, I would discuss around me to exchange and benefit from the various experiences and mutual reflections.

When you have detected your weaknesses, it is imperative to carry out personal work to improve your skills on these weak points.

This work can be done in parallel with Bootstrap, or out of sync.
For this, you will need to:

- find reliable resources,
- acquire knowledge,
- verify this knowledge concretely,
- test, hack to anticipate the consequences of your actions.

At this stage, the goal is not to become an expert, but to develop an intuition, which you will refine through practice.

CRUD

For the project, you will need an API to expose a CRUD to your web client.
It must expose, at the very minimum, the following routes:

- *item*/create
- *item*/read/:id
- *item*/update/:id
- *item*/delete/:id

Create/Read/Update/Delete in case you haven't made the connection with CRUD...

Read and understand the following code, as if you were to have to improve it.

Take some time to investigate the code snippets you do not master yet.

Modify it almost randomly trying to anticipate the consequences of your modification.
Once you feel at ease, add some routes, delete existing ones and rewrite them entirely.

AUTHENTICATION

You need a secure user authentication system.
A user must be able to register and identify himself.
This identifier must be used in the user interactions with your future product.

Such an authentication API should expose at least the following routes:

1. /signup
2. /login
3. /logout
4. /islogged

From the same code as the previous point, add these routes (in the very order they are introduced) and build a full authentication system.

DEPLOYEMENT

Deploy your API on Heroku.

Refer to your general documentation and look for the *How to deploy a web solution* guide to understand how a web deployment works.

POSTMAN

Postman is a very useful tool to test APIs.
Once you wrote your APIs, you will have to test them.

You should test successfully at least the following routes: entity create, entity read, entity update, entity delete, user signup, user login.

Install Postman, and check whatever internet references you fancy, to learn how to use it.
Once you created your first tests, check collections, Postman sandbox, Newman,...

Create procedural tests with Postman could be easy and save you time.

WEB CLIENT

When your API is ready and tested, build a nice and intuitive client that uses the totality of your routes.

Nothing more, nothing less.
Well, in fact, you can do more if you have time left.