

GENERAL INFORMATION

★ Guidelines

▶ Step 1: Build your database

▶ Step 2: Build the two routes

▶ Step 3: Create the corresponding views

📄 Deliverables

👤 Assessment grid

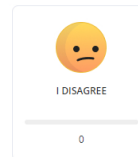
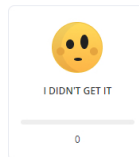
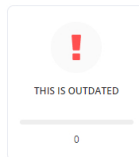
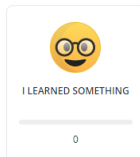
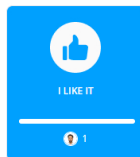
✅ Assessment grid

Here is the assessment grid that will be used for the peer review.

Don't hesitate to use it as you make progress on your project, to make sure you're going in the right direction!

Compétences	✅ Acquired
Build a database	<p>Creation of the database:</p> <ul style="list-style-type: none">☐ The learner created a MySQL database on the command line☐ The initialisation of the database is done from a file <p>Structure of the database:</p> <ul style="list-style-type: none">☐ The database contains a <i>schedules</i> table☐ The <i>schedules</i> table contains all required fields☐ All table fields have the appropriate type (following requirements) <p>Usage of the database:</p> <ul style="list-style-type: none">☐ The learner has linked his database to the application with the <i>mysql</i> module
Manage and display the content of a database	<p>Requests to the DB:</p> <ul style="list-style-type: none">☐ The learner uses a MySQL query (INSERT) to add a new schedule.☐ The learner uses a MySQL query (SELECT) to retrieve existing schedules. <p>Usage of the DB:</p> <ul style="list-style-type: none">☐ The data displayed on the site are the data present in the DB☐ When a new schedule is added, it appears on the <i>schedules</i> page.
Secure the use of a database	<p>Securing the connection to the DB:</p> <ul style="list-style-type: none">☐ The learner has protected the DB with a username and password☐ The connection to the DB is no longer possible if the password is changed. <p>Securing calls to the DB:</p> <ul style="list-style-type: none">☐ Arguments are passed to queries via "?" parameters.
Use JavaScript server side with Nodejs and Express	<ul style="list-style-type: none">☐ The learner used the ExpressJS framework <p>URL routing:</p> <ul style="list-style-type: none">☐ A route "/" displays the existing schedules☐ A route "/new" with the GET action displays the schedule creation form☐ A route "/new" with the POST action handles the input gathered by the schedule creation form <p>Views:</p> <ul style="list-style-type: none">☐ The learner set up a view engine to render HTML templates☐ The "/new" route with the GET action renders an HTML template that contains a form☐ The "/" route renders an HTML template that takes the schedules as parameter <ul style="list-style-type: none">☐ Toutes les fonctionnalités marchent sans erreur
Write code that is clear, concise and maintainable (practice)	<p>Presentation</p> <ul style="list-style-type: none">☐ The whole code is clean (example: indentation) and readable.☐ Comments are used sparingly to explain sensitive portions of code only. <p>Naming</p> <ul style="list-style-type: none">☐ The learner respected the "camelCase" naming convention, i.e. : <i>myFunction()</i>☐ All function and variable names are self-descriptive <p>Code organisation</p> <ul style="list-style-type: none">☐ The whole program is logically structured☐ The code is correctly factorised and does not repeat itself.☐ Each function has a single responsibility

React to this cheat sheet

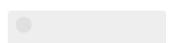
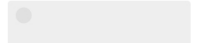
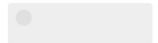


Thanks for your feedback! Lorcan will appreciate it.

FINISH



A question, a comment?
Post it in the forum!



Type your comment



Publish in the group: Incode (Aus) - 1

