**Weather Code Test**

In this test, you will create a **simple** Weather Forecast web app.

1. Get weather forecast by city name (using an API).
2. Show forecasts dynamically: type in a city name (E.g “Tel Aviv”), and show the Forecasts period, date/time, min temperature, max temperature, wind speed.
3. Create Forecast Save button. The save button will save only THE FIRST forecast item, meaning the most real-time item in the API’s response. (Note: if the forecast for a city already exists in the DB, update the forecast entry, do not keep a city duplication!)
4. Create a “load from db” button. The load button will load a forecast from the DB by the city’s name, the same one you saved in the last task.

On load show the city’s name, and replace the “period” in the title with “updated at” with the last update’s time. (see “examples” folder for clarification)

Required to use:

1. Openweathermap API - <http://api.openweathermap.org/data/2.5/forecast?q>={ CITY NAME }&units=metric&appid=e4b8b08c185638b825af37facfe1fabb

API Documentation -<http://openweathermap.org/forecast5>

1. Client Side – Use any framework or library you wish (or vanilla js)
2. Server Side – PHP + Laravel
3. DB – MySQL. (Note: Use your own migrations)

Setup:

1. Create a new laravel project, called weather
2. Create a new database (using PhpMyAdmin / DataGrip / the terminal / etc..)
3. Create a database migration, a model and a controller in Laravel
4. The migrations must have:

* id (increments)
* a UNIX timestamp (given by the API) (timestamp\_dt)
* city name (city\_name)
* minimum temperature (min\_tmp)
* maximum temperature (max\_tmp)
* Wind speed (wind\_spd)
* Laravel timestamps (updated\_at, created\_at)

Frontend code structure:

You can use any structure you see fit when it comes to the front end, either stored in the same project as laravel, using laravel views or in separate projects

Examples for the UI are included in the “examples” folder