Intellexi coding assignment (Laravel)

Technical requirements

## Introduction

The customer is a trail race organiser. They have been organising trail races for a couple of years and want a better way for participants to apply to their races and manage races, since they’ve only been using Excel spreadsheets.

Build a REST API for them using a small event driven CQRS microservice architecture. Since this kind of application will experience many more read than write requests, CQRS sounds like a good candidate.

Architecture

A white rectangular sign with black text

Description automatically generated

## API description

The API should enable the following operations:

* Create
* Patch
* Delete
* Get (one)
* Get (all)

Resources that require API implementation are Application and Race.

Users **do not** require API implementation and can be inserted directly into the database.

Application is defined by the following attributes:

* ID (uuid) required
* First name (255 characters) required
* Last name (255 characters) required
* Club (255 characters) optional
* Race ID (uuid) required

Application supports following API operations:

* GET (one)
* GET (all)
* CREATE
* DELETE

Race is defined by the following attributes – API required

* ID (uuid) required
* Name (255 characters) required
* Distance (5k | 10k | HalfMarathon | Marathon) required

Race supports following API operations:

* GET (one)
* GET (all)
* CREATE
* UPDATE
* DELETE

User is defined by the following attributes – API NOT required

* ID (uuid) required
* First name
* Last name
* E-mail
* DOB
* Role (Applicant | Administrator)

## Implementation

### DB

Create a database of your choice as part of the docker compose file.

### race\_application \_service

Laravel microservice implementing the mutating API endpoints (create, patch, delete). These API endpoints are only accessible to users with **Administrator** role. It updates its database (your choice, Cassandra or PostgreSQL)

What is required:

* API endpoints for CREATE, UPDATE, DELETE and GET (one/all) operations as described in the above section
* Authorization of the user by role
* As an Applicant, I get a list of races available to which I can apply
* As an Applicant, I select a race and get a form for application
* As an Applicant, I get a list of my current applications
* As an Applicant, I can delete my current applications
* As an Administrator, I get a list of races available to Applicants
* As an Administrator, I can add new races, update and remove existing ones
* Service needs to be included in the docker compose file

### Makefile

Add a makefile that will contain commands to run the development environment, build the application (PHP, docker) and run tests. Document the commands

## Guidelines

Pay attention to the following:

* Use composer
* Use JWT tokens for authentication/authorization
* Pay attention to separation of concerns principle
* Error handling
* API design
* Logging
* Configuration management (there should be a possibility to redefine configuration values)
* Tests are very important (unit, integration)
* Use docker compose to set up the development environment (database, messaging)

Expectations:

As a deliverable, we expect a GitHub repository (or any other git based repo) with the source

code. We would like the solution to contain clear instructions on how to set up and execute the project.

We expect the solution to be production ready.

Once you’re done with the solution, please push your code to the GitHub repository and provide a link with public access.

The task is complex and time consuming, so be prepared to highlight eventual gaps and areas which could be improved.