# Jamasoft Programming Challenge

## Overview

We would like you to build a proof-of-concept Restful API using Laravel, that functions as a web directory. A web directory is an online catalogue where various websites are listed, categorized and ranked based on user actions.

You might want to consider:

* Using starter kits or libraries such as Sanctum or Passport for a quick solution for authentication.
* If you struggle for time, focus on the underlying structure (migration, factories, controllers and their methods etc) even if that means some of the implementation for those is missing;

## User Requirements

Please use the following user stories as a list of functional requirements for your application.

There are three tiers of access to this system: users who are not logged in, users who are logged in and administrators.

### Unauthenticated User Requirements

* As an unauthenticated user I would like to have the directory’s websites presented to me in a categorized way, so content is sorted for me.
* As a user I would like to be able to search these websites so that I can quickly find content that is relevant to me based on a search term.

### Authenticated User

* As an authenticated user would like the ability to log in and log out so that the interactions I have with the system are identifiable and linked to me.
* As an authenticated user I would like to have the ability to submit my favourite website to the directory so that others can view them.
* As an authenticated user I would like to vote/unvote my favourite websites. Categories will then show them in order of how many votes they have so that the most relevant content is always at the top.

### Admin Requirements

* As an administrator I would like to be able to delete websites when needed.

## Technical Requirements

The following functional requirements should also be considered in your project:

* A submitted website can belong to multiple categories
* Users should be able to vote a website only once
* Search should work in combination with categories and ranking and be able to query 100 thousand records+ in the websites table in a reasonable time

## Want to go the extra mile?

* Seeders & Factories are a great plus
* Tests & doing TDD
* Postman collection to easily test the API

## What Matters

We will be looking to see the following in your code:

* **Production-grade** -As a rule of thumb, think how you would build this for a production system and write your code as such;
* **Efficiency** -your database migrations should consider structure, indexing and optimizations
* **Integration** -consider how a client would consume this API (potentially a JavaScript SPA)
* **Documentation** - your code should be easy to follow and easy to understand, with comments where needed
* **Scalable** - show us that your solution still works if it’s under heavy load or it has a lot of data. eg. does searching still work when your table has a few million rows? Do the queries still retrieve what they need to?

## How To Submit

Send a git repo link or a zip/rar archive containing your code and instructions on how to run it to [career@jamaconcept.com](mailto:career@jamaconcept.com)