**Kanban Services**

**Code and Architecture Explanation**

* All docker containers configurations are inside **‘build/’** directory and may find docker composer on root.
* On root **`.env`** file will contain all environment variables for containers, where we have database details.
* We have numerous options for frameworks to select, such as; Laravel, Symfony, Lumen and Slim but I have selected **Lumen** Micro-Framework per requirements for Kanban Services.
* Basic reason behind the selection of this framework is, This is lite and fast API micro-framework which is derived from Laravel, and has lots of features available as we have in Laravel Framework.
* Additionally, you easily may add all your libraries from Laravel framework, when ever you need to integrate this application with any other large scale application or wants to make this large.
* I have created fast and scalable **API endpoints** to facilitate **JSON** requests and to get json in response.
* Inside `src` directory, you may find API services code with framework.
* In API responses, I have also added proper **HTTP response codes** along with response data.
* We have endpoint route inside `**src/route/web.php**` directory, from where we may easily handle all routes for other new APIs. As we are designing APIs with version so I used `**v2/**` as a **prefix** with every endpoint. example: `**http://127.0.0.1/v2/project**`
* We have `**src/.env**` file which you may create different for every environments per requirements.
* Also I have added application configuration file in `**src/config/app.php**`, where I have defined all required configurations and roles in array and you also may add any other required parameters to modify and manage easily.
* You may find all controller inside `**src/app/Http/Controllers/**`, where we are submitting our requests and getting responses.
* For this application, I have added a **Facade Layer** and **Validation Layer**.
* **Facade Layer**, I have added this layer to write all business logics on a centralized place and to make things more simple by breaking up into small logical units. Which we may easily find and modify when needed. You may find all facade layers inside `**src/app/Facade/**` directory. Similarly, may find validators inside `**src/app/Validators/**`.