**Application Assumptions**

- The number of users generated will be based on a user input. The maximum users that can be generated for the external API is 5000 which is based on what's written on the documentation found here -> <https://randomuser.me/documentation>. The maximum users for the internal API is 100 as it is the number of users hard-coded in the database (using sqlite and populated using faker).

- Generated users will be represented on a table and can be sorted alphabetically based on their usernames in both ascending and descending orders.

- There are two tabs to switch between the external API and internal API. The html/css/javascript used to integrate this can be found here -> <https://www.w3schools.com/howto/howto_js_tabs.asp>

- Users generated on the internal API aren't given in random and has a set order.

**Application Instructions**

**-** The repositories *user\_generator* and *userGeneratorAPI* should be cloned locally. Both of the repositories can be found below:

*user\_generator*: <https://github.com/GerronT/user_generator>

*userGeneratorAPI*: <https://github.com/GerronT/userGeneratorAPI>

**-** Both applications require the node js libraries which can be installed by running **'npm install node'** in the command line for both project folders.

- Inside userGeneratorAPI/.env, make sure to set

DB\_CONNECTION=sqlite

and delete everything else that starts with DB\_\*.

- To run user\_generator, enter **'npm run serve'** in the command line while in its directory. To run userGeneratorAPI, enter **'php artisan serve'** in the command line while in its directory. The command line should show the links to their front-end in both cases.

- Users can be generated using the external API (randomuser.me) through user\_generator alone but switching to the internal API requires userGeneratorAPI to be running before users can be generated.