

# Consumer Manager

## (Laravel & Vue.js)

### Setup

1. Create a Laravel app from scratch, use version 5.6
2. Load Vue.js and [axios](#) (for AJAX requests)
3. Create table `consumers`.
  - The data you'll need for each consumer is: `id` (auto increment) , `name` , `age` , `city` .
  - Cities can be only Sofia, Plovdiv and Varna.
  - Create the table as a migration.
  - Use Laravel's Database Seeding to add 10 different records to the database (Check Restrictions paragraph)
  - Other table fields (created\_at, updated\_at, deleted\_at) are by your preference
4. Create a page that looks as the example below. Consumers listing and all actions on the page must be controlled via Vue.js

SEARCH			SORT OPTION ▾	
#	NAME	AGE	CITY	ACTIONS
1	Peter Petrov	32	Sofia	DELETE
2	Ivan Ivanov	17	Plovdiv	DELETE
3	Georgi Kirilov	71	Varna	DELETE
	NAME	AGE	CITY	ADD USER

### Listing

Pass the results from the database to your vue component (or the state, see Bonus p.1) in JSON format.  
User's names with age under 18 must be colored in **green** and above 60 must be colored in **red**.

### Actions

Search and Sort actions must not make AJAX requests it should be done in the front-end.

All actions happen without reloading the page.

- Search - a text input which filters the consumers list live while the user is typing. The search query must be matched against name, age and city at the same time.
- Sort - Order the consumers by the chosen column and direction. The options are:
  - ID Asc (default)
  - ID Desc
  - Name Asc
  - Name Desc
- Add - create a new record on the fly by sending a POST request. On add, the sorting and search are reset and the new record is added to the end of the list.
- Edit - edit a field by sending a PUT request. Only 1 field can be edited at a time! Edit mode for a field is activated when you double click on a field, then the text must become an input with ✓ and ✗ next to it. The ✓ will save the changes and ✗ will cancel them, original value before the edit began must be shown.
- Delete - remove a record by sending a DELETE request. The record is removed from the list.

## Restrictions

- Use backend validation
  - Name must be a string between 4 and 40 characters
  - Age must be an integer between 14 and 99
  - City must be Sofia, Plovdiv or Varna
- Use Eloquent query builder for touching the database
- Disable inputs/buttons while a request is being made - saving, deleting, etc.

## Bonus

Each bullet is a separate task. Can be done in any order.

1. Make use of Vuex for handling of the front-end state (similar to Flux and Redux)
2. Make the users page accessible only for logged in users
  - a. Use Laravel's Authentication
  - b. Consumers you are managing is different from users who can log in.
  - c. 1 user (administrator) is enough
3. Use less for the styling
4. Create another page `About`. It's content can be "Hello, this is the about page!"
  - a. Create Links on top of both pages. Use Blade to reuse the navigation
  - b. Use Vue Router to switch between the pages