Laravel / MINI CRM

Brainster Web Development Academy

Exercise

For this workshop we will create a project for managing companies and the employees working in each one. CRUDs for both companies and employees need to be created.

For each company we know:
name(required),
email (email, valid email address format),
logo (valid url),
website (valid url) and
country_id (required, foreign key to the countries table, value must exist in the companies table).

There is no need to create CRUD for countries. Instead, to create and populate the countries table, download the migration and the seeder from the following resource.
https://github.com/fabiofdsantos/laracountries/tree/master/database

In the create and edit forms for the company, the country field should be shown as dropdown with all the countries from the table available for selection.

Exercise

For each employee we know the
name (required),
email (required, valid email address format) and
phone (optional),
company_id (required, foreign key to the companies table, value must exist in the companies table).

In the create and edit forms for the employee, there shouldn't be a dropdown to select a company. Instead, in the companies HTML table, add a 'Manage employees` button that will get you to a page listing all the employees in that company. There should be a button `Add new employee`, after which employee create form is opened and newly created employee is automatically assigned to the selected company. To accomplish this, you can have the company_id in the url parameters. For example, the create employee form can be accessible on `/company/1/employee`, which means that we are adding new employee to the company with id: 1.

Use database migrations to create the required tables.
Use Laravel Requests to validate data when performing CRUD operations.

Implement search functionality in the view where all the companies are listed.
Implement search functionality in the view where all the companies are listed.

Exercise

Additionally, try to implement an authentication mechanism so that only signed in users can manage companies and employees.

For this, you will need to create a migration for storing users in database and logic for signing in. When the user signs in successfully, some value (i.e. 'user_id') is stored in session. All the routes for managing companies and employees should be accessible only for signed in users (users having the SESSION variable filled).

If someone is not signed in and attempts to access any of the routes for managing companies or employees, he should be redirected to the login page.

Since we have already done this with writing custom logic, although still not covered during the lessons, try to implement some of the built-in Laravel Authentication Starter Kits. Laravel Breeze is probably the easiest option, so you can get familiar with it.