Challenge #4 - Event Manager

The company "Events & brothers" wish to develop a functional prototype web application for manage events.

An event is composed by: Name, category (possible categories are: Conference, Seminar Congress or Course), place, address, start date, end date, and if the event is face or virtual.

For make a proof of concept, it is required to develop a web application for allow to the end user without start a session:

- Register to the website with email and password
- Start session with email and password (login page)
- Recover password

Once any user started a session, then the user can:

- List all events ordered descendant by creation date (from the most recent to the oldest).
- Each user can only see / manage his own events.
- See the detail of an event.
- Edit an event.
- Delete an event.

On other words make a CRUD (Create - Read - Update - Delete) for the event.

Additional Requirements:

Create a Profile Page for the registered user can setup his names, email, phone, photo, profile and login.

Admin Module:

Create a User Roles Module: Admin and regular user and the ability to assign users his profile for the admin

Manage Users: List / Add / Edit / Delete

Create a report for events: It list the events on a date range, the option to sort by any field / link to detail, export to csv file, and pagination for admin user.

Specifications:

1. The home page of the web application when the user not started a session is the start session page and when the user already started a session the page is the event list.
2. For this challenge use the PHP Sympony or Laravel Framework
3. For the manage of the authentication and authorization libraries of the framework are allowed
4. Not use libraries or tools for generate code for this challenge.
5. Use the front - end framework more useful for you (HTML basic / CSS /js, Bootstrap, Foundation, AngularJS, React, etc.)
6. Take the relevant decisions for analysis and design.

Additional Points:

JavaScript validations, graphic report pie chart, and publish on AWS instance

Use git for deliver the code, clean code and refactor.

Deliver en ER model, class diagram and running solution.

Symfony Links:

https://symfony.com/what-is-symfony

https://symfony.com/doc/current/index.html

https://symfony.com/doc/current/introduction/from_flat_php_to_symfony2.htm

Laravel Links:

https://laravel.com/

https://laravel.com/docs/5.6#installation