**Digital Balance Challenge**

**Laravel7 (Auth+Policy+Sortable+storage(S3) ) ,Docker and AWS S3**

I used App\Training model to store the metadata of the training.

And AWS S3 to host securely and durably the actual documents (videos,Audio,Images, PDF…..).

The secure document in AWS S3 includes:

* Server side encryption.
* Document Versioning.
* Multifactor Delete Protection.
* IAM role Only Laravel Admin user has full access on S3.

1. **Infrastructure**

As infrastructure I used an official Docker Composer for Laravel 7 <https://hub.docker.com/r/bitnami/laravel>

Which contains two containers one for laravel 7 and one for mysql

I customise the docker-compose.yml by adding 2 docker volumes as persistence.

One for the laravel php source code and one for mysql database as following:

Mapping Docker host volume: ../database to the mysql container :/bitnami/mariadb/data

volumes:

     - ../database:/bitnami/mariadb/data

Mapping Docker host volume: ./(the current dirrectory) to the Laravel container :/app

volumes:

      - ./:/app

I put the file docker-compose.yml in git hub <https://github.com/YCHOUGRANI/DigitalBalance_laravel_test>

For the deployment:

1. Install Docker and Docker compose:

<https://docs.docker.com/engine/install/>

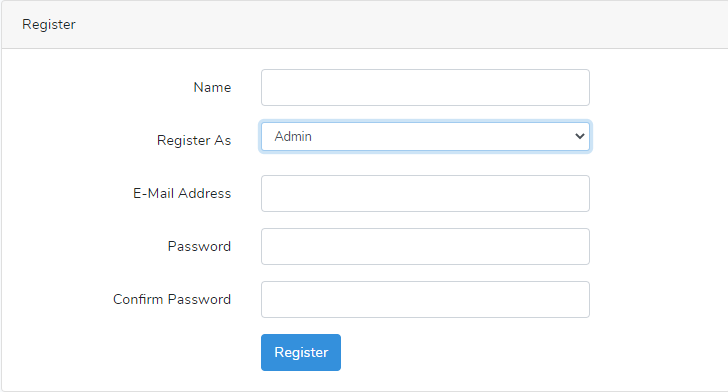
<https://docs.docker.com/compose/install/>

1. Then clone the source code from git <https://github.com/YCHOUGRANI/DigitalBalance_laravel_test>
2. Create a folder with name “database” in the same level as DigitalBalance\_laravel\_test folder.
3. Run Docker-composer up.
4. Docker-compose exec myapp php artisan migrate:refresh --seed
5. Browse the link <http://localhost:3000/>
6. **Registration Form**

As the behaviour of the application depends on the role of the user.

I customise the registration form by adding a list box for the role. (Admin,Paid Subscriber,Free Subscriber) This is just for the illustration, in real word you can use different group Staff ,Teacher, Writer…etc

1. Admin: Can create,delete, update trainings.
2. Paid subscriber: Can search and play the media (video,audio,picture,pdf…..).
3. Free Subscriber: Can only search and see only the title, the type and the description of an available training and not the document itself.



1. **Training List Page:**

The home page contains a list of available training:

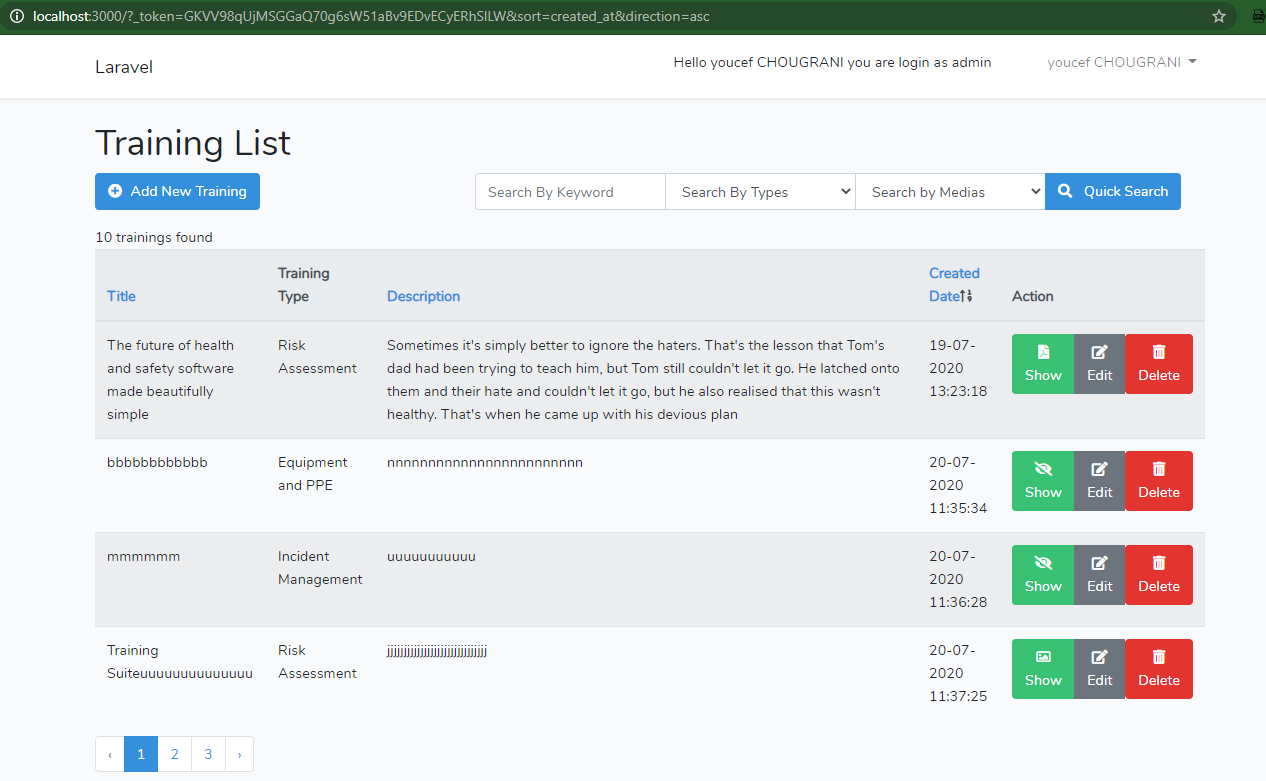
As I’m login as admin so I can Add, Delete and Update training.(I used Laravel Policy /app/Policies/TrainingPolicy.php)

Example policy for “Add New Training” Button:

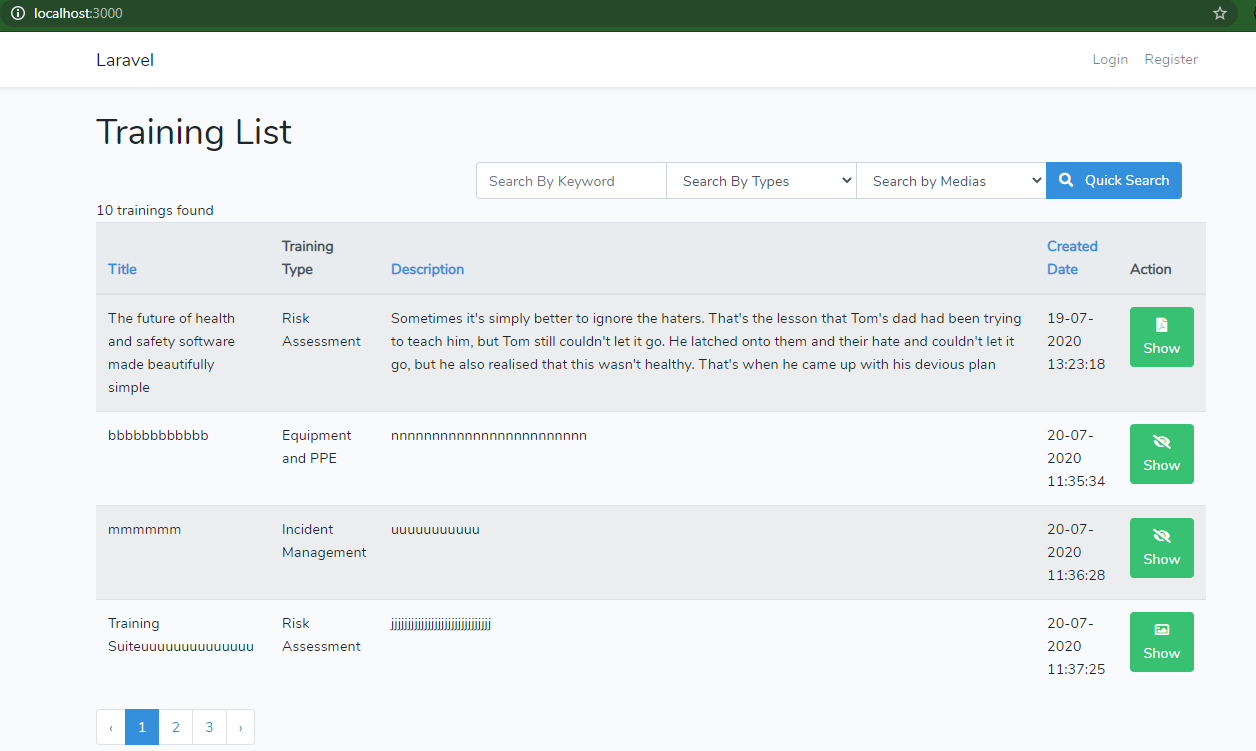
@can('create',\App\Training::class)

                <a class="btn btn-primary" href="{{route('training\_create')}}"><i class="fa fa-plus-circle mr-2"></i>Add New Training</a>

       @endcan



Here an example for a non-login user (guest user), he can only search and see the metadata of training.



The show icon inside the Action column is dynamically changing according to the mime type of the document.

I did it by creating custom field inside the Model \App\Training and I used filter collection.

The key of the collection is the name of the font awesome class.

public function getIconAttribute()

    {

        $ext=$this->attributes['extension'];

        $medias\_map\_collection= collect([ 'fas fa-file-pdf' => ['pdf','txt'],

        'fas fa-video' => ['mov','mp4','mpg','mpeg','ogv','webm'],

        'fas fa-volume-up' => ['wav','mp3'],

        'far fa-image' => ['jpeg','png','bmp','gif','svg','jpg']

        ]);

        $filtered = $medias\_map\_collection->filter(function ($value, $key) use ($ext) {

                   $value\_collection=collect($value);

               return $value\_collection->contains($ext);

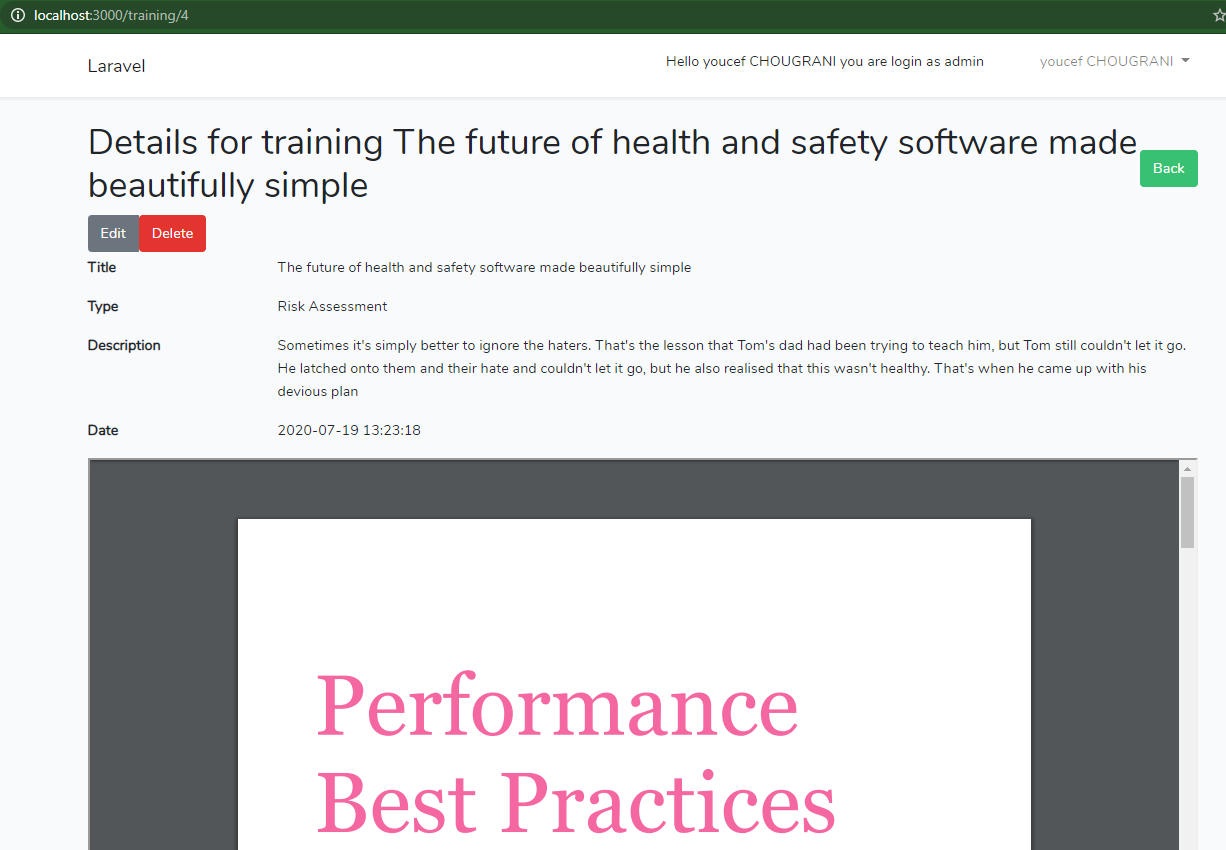
        });

        return $filtered->keys()->last();

    }

1. **Training show page.**

This is an example of showing pdf document in Laravel but the document is hosting securely in AWS S3 (only Laravel Admin user has full access on S3)



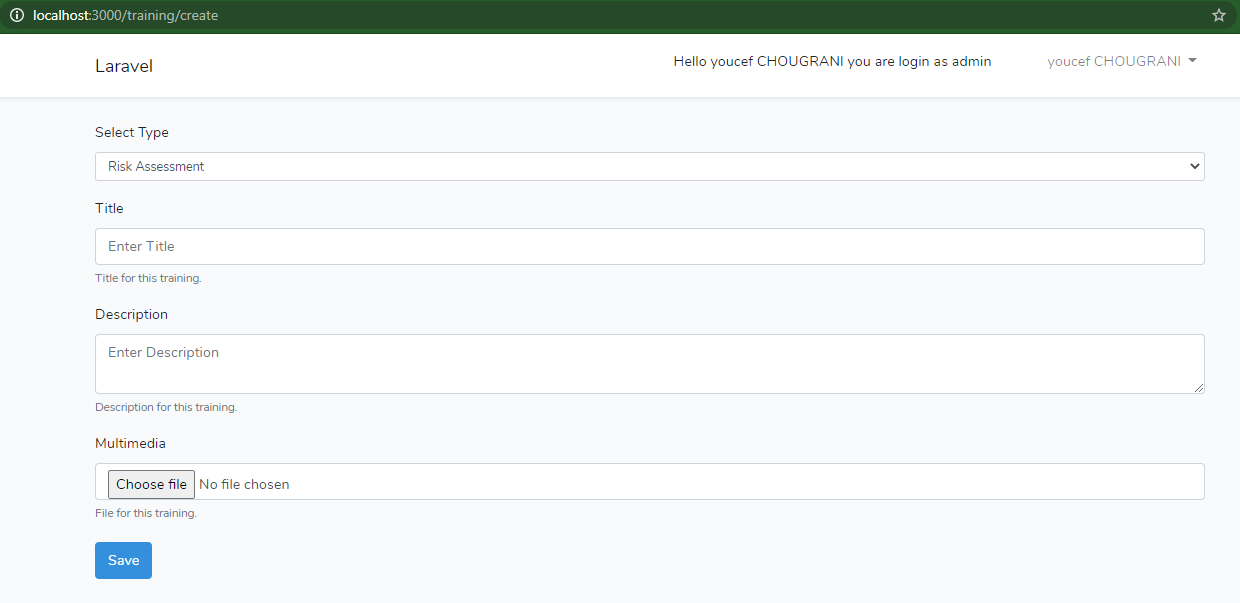
1. **Training Create Form**

After fill in the form and click on save button, the system store the metadata of the training into mysql table and the document is stored in AWS S3 on the following bucket:

|  |  |
| --- | --- |
|  | [https://dblx-training.s3.eu-west-2.amazonaws.com](https://dblx-training.s3.eu-west-2.amazonaws.com/images/s6ERpucDAUhJiJlxUljo9Wrj69h5k0ZsFhrIfmVi.pdf) |
|  | This is the config in .env file:  AWS\_ACCESS\_KEY\_ID=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  AWS\_SECRET\_ACCESS\_KEY=\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  AWS\_DEFAULT\_REGION=eu-west-2  AWS\_BUCKET=dblx-training  AWS\_URL=https://dblx-training.s3.eu-west-2.amazonaws.com |

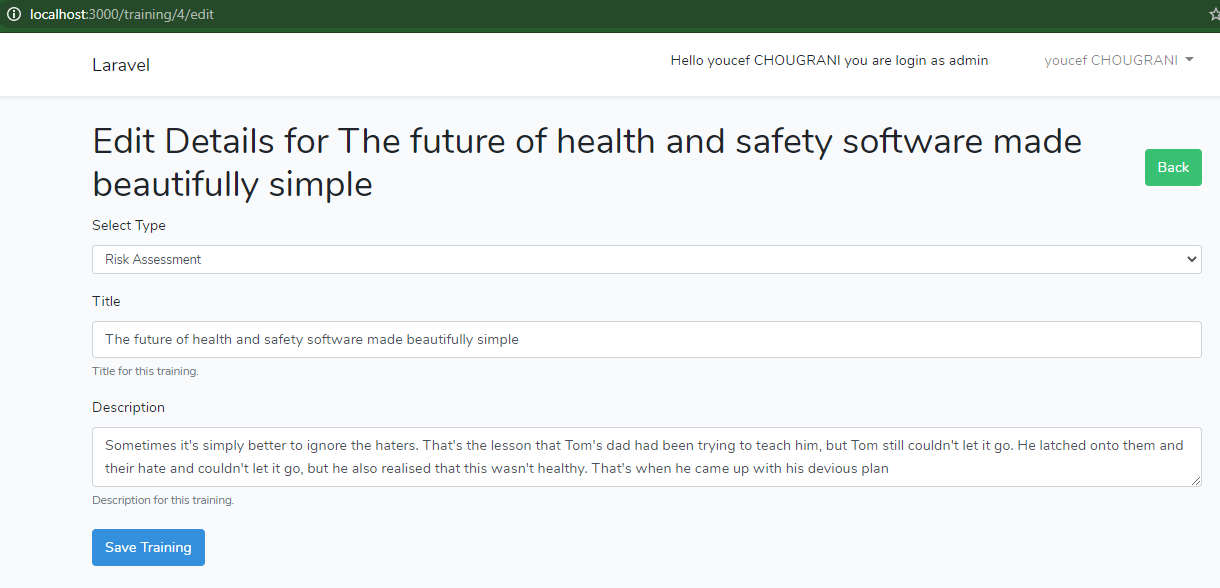
For laravel I used the following package

composer require league/flysystem-aws-s3-v3



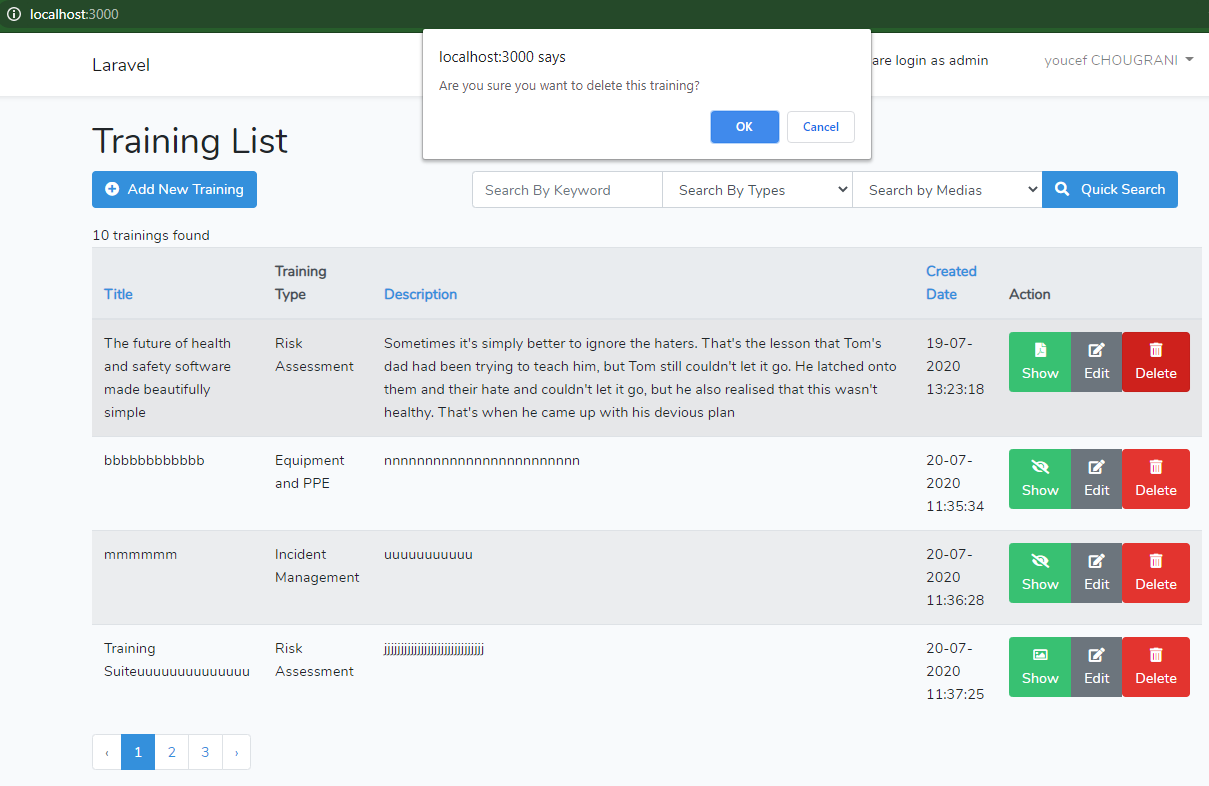
1. **Training Edit Form**

The admin can update only the title, the type and the description.



1. **Training Delete Form**

The delete button enable the admin user to delete both the metadata from the mysql table and also from AWS S3.

****