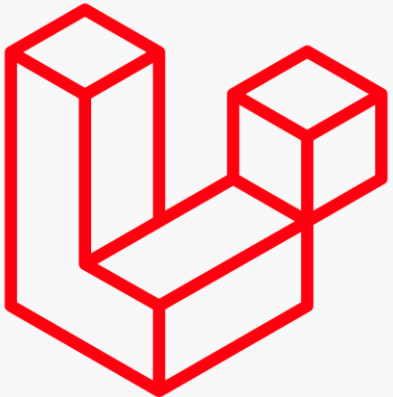


Laravel Training d3

Tarmizi Sanusi

Khirulnizam Abd Rahman

<http://fstm.kuis.edu.my/blog/laravel>



Module Outcome

Day 3

- Alert using Session
- File Storage
- Integrating Bootstrap Themes from startbootstrap
- Search Method

2. Route to appropriate Laravel Controller

1. Submit User Request



Routing

routes/web.php

app/Http/Controllers

Controller

3. Interact with Data Model

4. Controller invokes results View

resources/views

View

5. Render view in users browser

Model

app/



Database

Alert Using Session

When you retrieve an item from the session, you may also pass a default value as the second argument to the get method.

<https://github.com/samtarmizi/laravel-spr/commit/43ca153eebb7692ed5d430f2d9e3c2ccc49ab3f6>

index.blade.php

```
@if (session()->has('alert'))
    <div class="alert {{ session()->get('alert-type')}}">
        {{session()->get('alert')}}
    </div>
@endif
```

TrainingController.php

```
return redirect()
    ->route('training:index')
    ->with(['alert-type' => 'alert-primary','alert'=> 'Your training saved']);
```

```
return redirect()
    ->route('training:index')
    ->with(['alert-type' => 'alert-success','alert'=> 'Your training updated']);
```

```
return redirect()
    ->route('training:index')
    ->with(['alert-type' => 'alert-danger','alert'=> 'Your training deleted']);
```

File Storage

1. Create migrations to alter trainings table
2. Add attachment input in views
3. Create Symbolic Link
4. Check in Controller if the input has attachment and update row with the file name.
5. Delete the file on storage if the row will be deleted
6. Open attachment file

File Storage

1. Create migrations to alter table

```
php artisan make:migration alter_articles_table_add_attachment
```

```
class AlterTrainingsTableAddAttachment extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::table('trainings', function (Blueprint $table) {
            $table->string('attachment')->nullable()->after('trainer');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::table('trainings', function (Blueprint $table) {
            $table->dropColumn('attachment');
        });
    }
}
```

File Storage

2. Add attachment input in views

```
<form action="{{ route('training:store') }}" method="POST" enctype="multipart/form-data">
    @csrf
    <div class="form-group">
        <label for="title">Title</label>

        <label for="title">Trainer</label>
        <input type="text" class="form-control" name="trainer">
    </div>
    <div class="form-group">
        <label for="title">Attachment</label>
        <input type="file" class="form-control" name="attachment">
    </div>
    .. .. ..
```

create.blade.php

File Storage

3. Create Symbolic Link

The public disk is intended for files that are going to be publicly accessible.

By default, the public disk uses the local driver and stores these files in `storage/app/public`.

To make them accessible from the web, you should create a symbolic link from **public/storage** to **storage/app/public**.

To create the symbolic link, you may use the `storage:link` Artisan command:

```
php artisan storage:link
```


File Storage

4. Check in Controller if the input has attachment and update row with the file name.

In Model, update *fillable* with **attachment** to allow mass assignment method.

```
protected $fillable = [  
    'title', 'description', 'trainer', 'user_id',  
    'title', 'description', 'trainer', 'user_id', 'attachment'  
];
```

```
-41,6 +43,13 @@ public function store(Request $request)
```

```
    $user = auth()->user();  
    $training = $user->trainings()->create($request->only('title','description','trainer'));
```

```
    if($request->hasFile('attachment'))  
    {  
        $filename = $training->id.'-'. date("Y-m-d").'.'.$request->attachment->getClientOriginalExtension();  
        Storage::disk('public')->put($filename, File::get($request->attachment));  
        $training->update(['attachment' => $filename]);  
    }
```

```
    return redirect()  
        ->route('training:index')  
        ->with(['alert-type' => 'alert-primary','alert'=> 'Your training saved']);
```

File Storage

5. Delete the file on storage if the row will be deleted.

```
if($training->attachment != null){  
    Storage::disk('public')->delete($training->attachment);  
}
```

TrainingController.php on delete method

File Storage

6. Open attachment

```
<div class="form-group">
  <label for="title">Attachment</label>
  <a href="{{ env('APP_URL') }}/storage/{{ $training->attachment }}" target="_blank">Open Attachment</a>
</div>
```

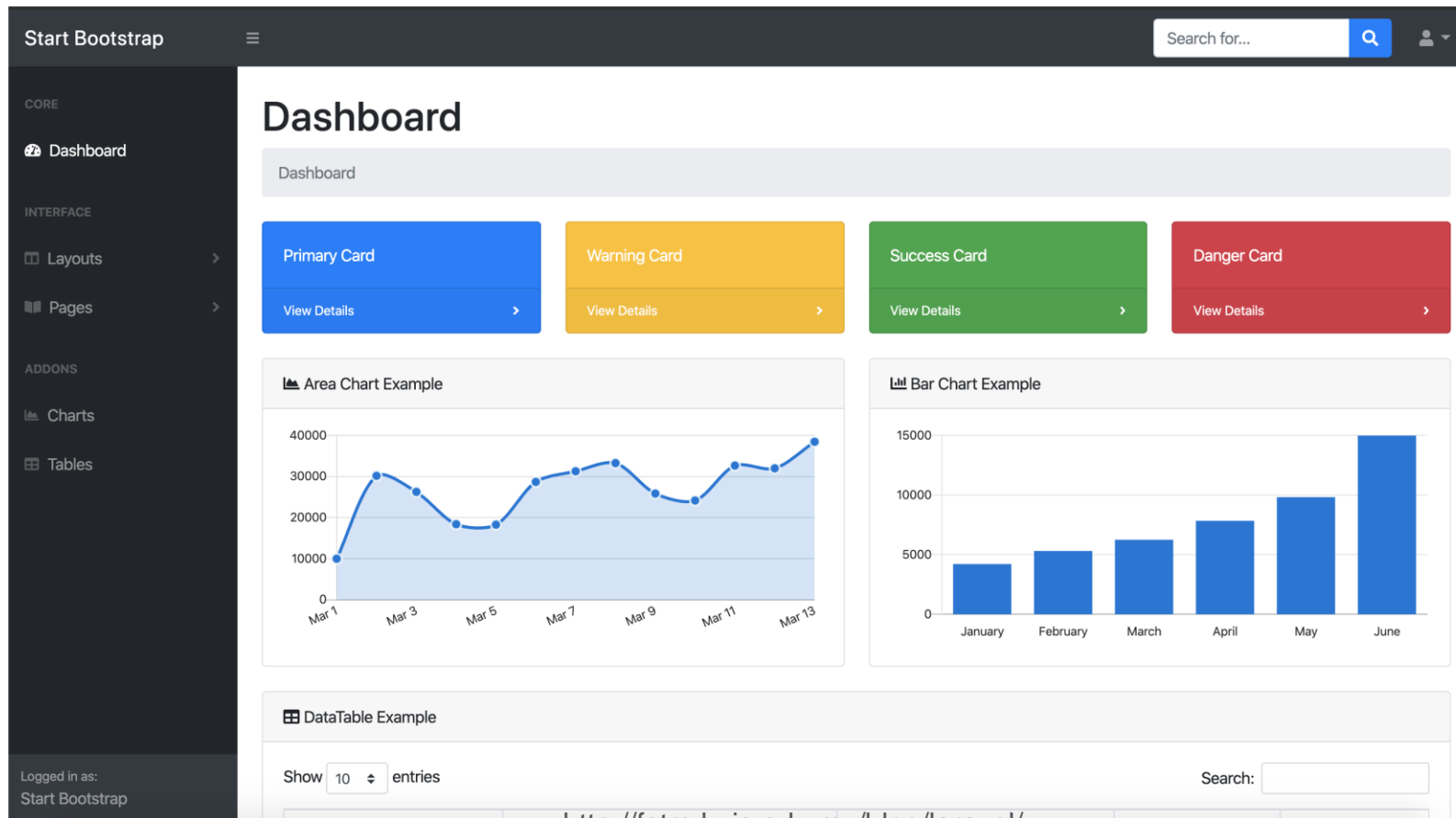
show.blade.php

Hacks

File Storage- <https://github.com/samtarmizi/laravel-spr/commit/fa63ab5c3074cae15ead8bd33bab530c879e39b2>

Choosing a Templates

<https://startbootstrap.com/templates/sb-admin/>



<http://fstm.kuis.edu.my/blog/laravel/>

CORE

[🏠 Dashboard](#)

INTERFACE

[🏠 Layouts](#) >[📄 Pages](#) >

ADDONS

[📊 Charts](#)[📄 Tables](#)Logged in as:
Start Bootstrap

Dashboard

Dashboard

Primary Card

View Details >

Warning Card

View Details >

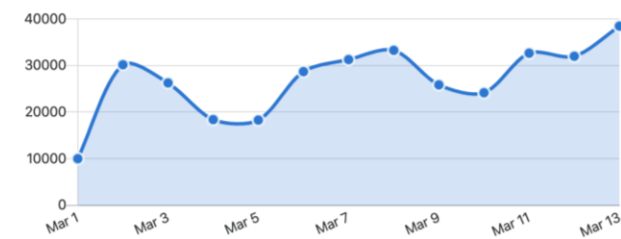
Success Card

View Details >

Danger Card

View Details >

📊 Area Chart Example



📊 Bar Chart Example



Integrate Bootstrap with Blade templating

Integrate with Bootstrap template

- Create a folder on **resources/views** named **admin**
- Create a folder on **resources/views/admin** named **includes** and **layouts**
- Create files on **resources/views/admin/includes** named:

footer.blade.php

sidebar.blade.php

navbar.blade.php

- Create files on **resources/views/admin/layouts** named:

main.blade.php

Adding Assets

- Move all assets to public path.
- Copy **index.html** to **resources/views/admin/layouts/main.blade.php**
- Change local call assets using asset

From: `<script src="js/scripts.js"></script>`

To: `<script src="{!! asset('js/scripts.js') !!}"></script>`

1. Find <footer> to </footer> tag then cut and paste to footer.blade.php
put @include('admin.includes.footer') on the cut area
2. Find <navbar> to </navbar> with top navbar class then cut and paste to navbar.blade.php
put @include('admin.includes.navbar') on the cut area
3. Find <navbar> to </navbar> with sidebarclass then cut and paste to sidebar.blade.php
put @include('admin.includes.sidebar') on the cut area
4. Find <main> to </main> tag and **replace the content only** with @yield('content')

Go to resources/views/trainings/

Changes on all blade file

From @extends('layouts.app')

To @extends('admin.layouts.main')

<https://github.com/samtarmizi/laravel-spr/commit/059077d9f11e76b129fee88ff702f7fb96ea093e#diff-137a2c14c3be2030f72ccc066194a803>

Update NavBar

resources/views/admin/includes/navbar.blade.php

```
@
'sb-topnav navbar navbar-expand navbar-dark bg-dark">
;="navbar-brand" href="index.html">Start Bootstrap</a><button class="btn btn-link btn-sm order-1 order-lg-0" id="side
;="navbar-brand" href="{{ url('/') }}">{{ config('app.name', 'Laravel') }}</a><button class="btn btn-link btn-sm orde
ivbar Search-->
.ass="d-none d-md-inline-block form-inline ml-auto mr-0 mr-md-3 my-2 my-md-0">
/ class="input-group">
.5 @@
<div class="dropdown-menu dropdown-menu-right" aria-labelledby="userDropdown">
    <a class="dropdown-item" href="#">Settings</a><a class="dropdown-item" href="#">Activity Log</a>
    <div class="dropdown-divider"></div>
    <a class="dropdown-item" href="login.html">Logout</a>
    <a class="dropdown-item" href="{{ route('logout') }}"
        onclick="event.preventDefault();
            document.getElementById('logout-form').submit();">
        {{ __('Logout') }}
    </a>

    <form id="logout-form" action="{{ route('logout') }}" method="POST" style="display: none;">
        @csrf
    </form>
</div>
```

Update Sidebar

resources/views/admin/includes/sidebar.blade.php 

@@ -50,6 +50,6 @@

</div>

<div class="sb-sidenav-footer">

<div class="small">Logged in as:</div>

- Start Bootstrap

+ {{ Auth::user()->name }}

</div>

Hacks

Update Navbar and Sidebar:

<https://github.com/samtarmizi/laravel-spr/commit/64c5705255df0b4d55ae95fe20194b3eb4a1af61>

Search

1. Create form to input keyword
2. Query the keyword that match with title

Training Index			
		<input type="text" value="laravel"/>	<input type="button" value="Q"/> <input type="button" value="+ Create New Training"/>
ID	Title	Creator	Actions
13	Laravel Training 6 Days(Advanced)	Prof. Francesco Haag IV	<input type="button" value="Show"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>
18	Laravel Medkad	Vincent Harding	<input type="button" value="Show"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/>

127.0.0.1:8000/trainings?keyword=laravel

Search - Form

Using GET method to
named route **training:index**

```
<div class="card-header">Training Index
  <div class="float-right">
    <form
      class="d-none d-md-inline-block form-inline ml-auto mr-0 mr-md-3 my-2 my-md-0"
      method="GET"
      action="{{ route('training:index') }}"
    >
      <div class="input-group">
        <input
          class="form-control"
          type="text" placeholder="Search for..."
          aria-label="Search" aria-describedby="basic-addon2"
          name="keyword"
          value="{{ request()->get('keyword') }}"
        />
        <div class="input-group-append">
          <button class="btn btn-primary" type="submit"><i class="fas fa-search"></i></button>
        </div>
      </div>
    </form>
    <a href="{{ route('training:create') }}" class="btn btn-primary">+ Create New Training </a>
  </div>
</div>
```

Search – Index Method

Check if there is keyword, if there is keyword on query string, run query to search column where like title.

If no keyword, the method will return collection rows from trainings table.

```
public function index(Request $request)
{
    if($request->get('keyword') != null)
    {
        $keyword = $request->get('keyword');
        $trainings = Training::where('title','LIKE','%'.$keyword.'%')->get();
    }
    else
    {
        $trainings = Training::all();
    }
    return view('trainings.index')->with(compact('trainings'));
}
```


Search - Pagination

The paginate method automatically takes care of setting the proper limit and offset based on the current page being viewed by the user and append with keywords.

resources/views/trainings/index.blade.php

```
@@ -54,6 +54,8 @@  
  
                @endforeach  
            </tbody>  
        </table>  
+        {{-- {{ $trainings->links() }} --}}  
+        {{ $trainings->appends(['keyword' => request()->get('keyword')])->links() }}
```

app/Http/Controllers/TrainingController.php

```
@@ -13,11 +13,11 @@ public function index(Request $request)  
  
    if($request->get('keyword') != null)  
    {  
        $keyword = $request->get('keyword');  
        $trainings = Training::where('title','LIKE','%'.$keyword.'%')->get();  
        $trainings = Training::where('title','LIKE','%'.$keyword.'%')->paginate(5);  
    }  
    else  
    {  
        $trainings = Training::all();  
        $trainings = Training::paginate(5);  
    }  
}
```

<https://github.com/samtarmizi/laravel-spr/commit/6b0d8b3c64bd42c58af97fc49b2c48d983273f1d>

<http://fstm.kuis.edu.my/blog/laravel/>