



HELP TEACHER

Documentación técnica

Isabel Rodriguez
rodpeisa@gmail.com

Introducción:

El proyecto se compone por la creación de informes individualizados, en los que los alumnos en un módulo concreto tienen un informe el cual se les puede enviar por correo.

Para llegar a ello se crea un índice donde se encuentran todos los alumnos del centro, se puede hacer una búsqueda por el nombre del alumno.

Este alumno se puede editar, también se pueden crear nuevos alumnos, al hacerlo hay que rellenar los campos de su información personal, al hacerlo te redirige automáticamente a la segunda parte donde seleccionas el grupo en el que está y los módulos a los que asistirá, al igual que la fecha en la que se matriculó.

Tras ello te redirige a la página principal, donde aparecerá ya el nuevo alumno.

Al seleccionar el nombre de un alumno se muestran todos sus datos (los que hay en la principal y algunos más), así como los módulos en los que este matriculado.

Se puede editar los módulos en los que este matriculado actualmente y poner si ha repetido curso o no.

Al seleccionar un módulo en concreto te muestra el informe, de no haber un informe te muestra la opción de crearlo, de haberlo, la opción de editarlo y de enviarlo al alumno.

A parte de esta opción principal, te permite gestionar a los profesores y crear nuevos profesores con sus contraseñas de usuario para acceder a la plataforma.

¿Qué se ha utilizado?

Para la creación de la aplicación web se ha utilizado el lenguaje de programación PHP con el framework de Laravel, se ha utilizado también una base de datos mysql gestionada con phpmyadmin.

Código específico.

Enviar emails

ruta: /routes/web.php

Que se hace: se mandan los datos del informe por url, luego se añaden en un array y este se pasa a la función mail, que es la encargada de enviar el mensaje con los datos guardados en el array colocados debidamente y según la estructura del fichero de envío.

Fichero de envío ruta: /resources/views/alumnos/enviar.blade.php

```
Route::get('enviar/{email}/{id_modulo}/{id}/{valoracion}/{apreciacion}/{actividades}/{evaluacion}',function($email,$id_modulo,$id,$valoracion,$apreciacion,$actividades,$evaluacion){

    $data = array('email' => $email, 'id_modulo'=>$id_modulo, 'valoracion'=>$valoracion,
        'apreciacion'=>$apreciacion,'actividades'=>$actividades,'evaluacion'=>$evaluacion); // Empty array

    Mail::send('alumnos.enviar',$data, function($message) use ($data){

        $message->from('rodpeisa@gmail.com','Isa');
        $message->to($data['email'])->subject('Informe del Modulo '.$data['id_modulo']);

    });

    return redirect()->back()->with('message', 'El informe a sido enviado');

});
```

```

<?php
<html OCTYPE html>
<html lang="es">
<head>
<meta charset="utf-8">
</head>
<body>
<h1>Informe</h1>
<p>VALORACIÓN DEL APRENDIZAJE REALIZADO: <?php echo $valoracion ?> </p>
<p>APRECIACIÓN DEL GRADO DE CONSECUCIÓN DE LAS CAPACIDADES ENUNCIADAS EN LOS MÓDULOS PROFESIONALES QUE
HAN DE SER OBJETOS DE RECUPERACIÓN: <?php echo $apreciacion ?></p>
<p>ASIGNACIÓN DE ACTIVIDADES DE RECUPERACIÓN AL ALUMNADO Y, EN SU CASO, APLICACIÓN DE MEDIDAS EDUCATIVAS
ESPECIALES: <?php echo $actividades ?></p>
<p>INDICACIÓN EXPRESA DE LA EVALUACIÓN FINAL EN QUE SERÁN EVALUADOS: <?php echo $evaluacion ?></p>
</body>
</html>

```

Generar alumno matriculado:

/resources/views/alumnos/create.blade.php

En esta ruta hace que se muestre el formulario para añadir los datos personales del alumno, una vez rellenados te manda los datos al controlador.

/app/Http/Controllers/AlumnoController.php

```

/*
 *
 */
public function store(Request $request)

{
    $ca=Alumno::find($request['id']);
    $c=Alumno::create($request->all());
    return redirect('matriculas/create/'.$request['id']);
}

```

En esta parte del controlador selecciona el alumno que se acaba de crear y te mando sus datos a la nueva vista donde se creara la matricula del alumno.

/resources/views/matriculas/create.blade.php

En esta vista habrá un formulario donde seleccionaras a que grupo pertenecerá el alumno, una vez seleccionado manda ese dato a la siguiente vista.

/resources/views/matriculas/create2.blade.php

En la segunda vista de la matricula según el grupo seleccionado elegirás las materias en las que estará matriculado el alumno y mandara los datos al controlador.

/app/Http/Controllers/MatriculaController.php

```

    public function store(Request $request)
    {
        $c = $request->input('modulos_ids_originales');
        $c = implode(',', $c);
        $input = $request->except('modulos_ids_originales');
        //Assign the "mutated" news value to $input
        $input['modulos_ids_originales'] = $c;
        Matricula::create($input);
        return redirect()->route('alumnos.index');
    }

```

En este controlador añade los módulos seleccionados en un array y los junta con el resto de los datos para crear la matricula, ¿Por qué se hace esto así? Porque en la base de datos que se nos proporciono los id de los módulos venían juntos en una cadena separados por comas, por lo que es necesario juntar los módulos de esa forma para añadirlos a la base de datos.

Acceso solo a logueados y admins

Para restringir el acceso solo a admins se creó el middleware siguiente:

```

AdminMiddleware x
<?php
namespace App\Http\Middleware;
use Closure;
class AdminMiddleware
{
    /**
     * Handle an incoming request. User must be logged in to do admin check.
     *
     * @param \Illuminate\Http\Request $request
     * @param \Closure $next
     *
     * @return mixed
     */
    public function handle($request, Closure $next)
    {
        if (\Auth::user()->is_admin) {
            return $next($request);
        }

        return redirect()->guest('/home');
    }
}

```

Y a logueados este:

```
<?php

namespace App\Http\Middleware;

use Illuminate\Auth\Middleware\Authenticate as Middleware;

class Authenticate extends Middleware
{
    /**
     * Get the path the user should be redirected to when they are not authenticated.
     *
     * @param  \Illuminate\Http\Request  $request
     * @return string
     */
    protected function redirectTo($request)
    {
        if (! $request->expectsJson()) {
            return route('login');
        }
    }
}
```

Este middleware se añade al controlador o la ruta de la vista para restringir el acceso:

```
public function __construct()
{
    $this->middleware('auth');
    $this->middleware('admin');
}

//...

Route::get('/modulos/create/{id}', 'ModuloController@create');
Route::resource('/informes', 'InformeController')->middleware('auth');
Route::get('/informes/create/{id}/{id_modulo}', 'InformeController@create')->middleware('auth');
```