

Tasca M10

Problema:

Descripcion

Solucion:

Descripció

Crear una aplicació, per a gestionar el fons de una llibreria tècnica, aplicant el patró de disseny de software MVC(Model-Vista-Controlador).

Nivell 1

- Exercici 1

Definir les rutes principals que tindrà el nostre lloc web. Per a començar simplement indicarem que les rutes retornen una cadena (així podrem comprovar que s'han creat correctament). A continuació s'inclou una taula amb les rutes a definir (Configura els mètodes GET, POST, UPDATE o DELETE d'acord a el tipus de ruta).

Ruta	Text a mostrar
/	Pantalla principal
login	Login usuari
logout	Logout usuari
catalog	Llista llibres
catalog/show/{id}	Vista detall llibre {id}
catalog/create	Afegir llibre
catalog/edit/{id}	Modificar llibre {id}

- Exercici 2

Crea les vistes php i associa-la a cadascuna de las rutes següents seguint la següent estructura:

Vista	Carpeta	Ruta associada
home.blade.php	resources/views/	/
login.blade.php	resources/views/auth/	login
index.blade.php	resources/views/catalog/	catalog
show.blade.php	resources/views/catalog/	catalog/show/{id}
create.blade.php	resources/views/catalog/	catalog/create
edit.blade.php	resources/views/catalog/	catalog/edit/{id}

Crea una col·lecció a [Postman](#) i guarda la petició del punt anterior en ella. Exporta la col·lecció i guarda-la en un directori nou del projecte.

- Exercici 3

Defineix un middleware i registreu-lo globalment a nivell de projecte. Aquest middleware serà l'encarregat d'imprimir la data actual a la pantalla.

Crear els formularis necessaris per a poder realitzar un CRUD sobre el sistema de gestió de la biblioteca tècnica. Heu de validar que la informació introduïda a l'usuari sigui correcta.

Nota: No cal crear la lògica per treballar amb la BD.

Nivell 2

- Exercici 4

Creeu un middleeare a nivell de controlador. Aquest middleware serà l'encarregat d'imprimir la data actual a la pantalla abans de carregar la informació.

- Exercici 5

Crea un sistema que gestiona l'error 404 a nivell de projecte. Podeu utilitzar la funció Resposta i la redirecció de Laravel.

Nivell 3

- Exercici 6

Afegeix la vista d'inici de sessió, registre i recuperació de contrasenya que l'usuari necessita per accedir a l'aplicació.

Nota: No cal crear la lògica per treballar amb la BD.

- Exercici 7

Agafeu les dades d'inici de sessió al sistema de galetes de Laravel. En cadascuna de les vistes has d'incrustar un navbar en el qual apareix el nom d'usuari registrat

Solucion:

1.- Rutas del proyecto:

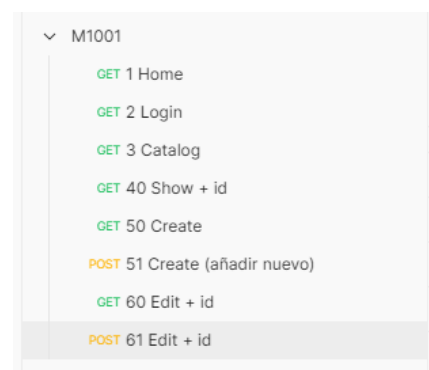
Estas son las rutas que se han generado para el proyecto.

Metodo	Uri:	Name:	Action:	Middleware :
GET	/		App\Http\Controllers\libreriaController@home	mostrarfecha
GET	catalog	catalog	App\Http\Controllers\libreriaController@catalo	mostrarfecha
GET	catalog	catalog	App\Http\Controllers\libreriaController@catalog	mosfechatrar
GET	catalog/create	catalog.create	App\Http\Controllers\libreriaController@create	mosfechatrar
POST	catalog/create	catalog.createStore	App\Http\Controllers\libreriaController@createStore	mosfechatrar
DELETE	catalog/delete/{id}	catalog.destroy	App\Http\Controllers\libreriaController@destroy	mosfechatrar
PUT	catalog/edit	catalog.editStore	App\Http\Controllers\libreriaController@editStore	mosfechatrar
GET	catalog/edit/{id}	catalog.edit	App\Http\Controllers\libreriaController@edit	mosfechatrar
GET	catalog/show/{id}	catalog.show	App\Http\Controllers\libreriaController@show	mosfechatrar

Ademas se añaden todas las rutas creadas por auth.

2.- Postman:

Se crea la colección de Postman.



Se ubica en el fichero M1001/Postman/**M1001.postman collection.json**

Se crean todo los petodos y en edit y create se añade el CSRF, para que puedan funcionar. Sera necesario cambiar este campo cuando se realizen las pruebas.

3.- Middleware:

Se crea el middleware para mostrar la fecha, de dos formas diferentes: mostrando la fecha por php, y utilizando la fecha de Laravel helper.

Se crea el Middleware:

php artisan make:middleware showDate y se crea:

M1001/app/Http/Middleware/ShowDate.php

Se registra en app/Http/Middleware/kernel.php

'mostrarfecha' => \App\Http\Middleware\ShowDate::class

Y se añade el código:

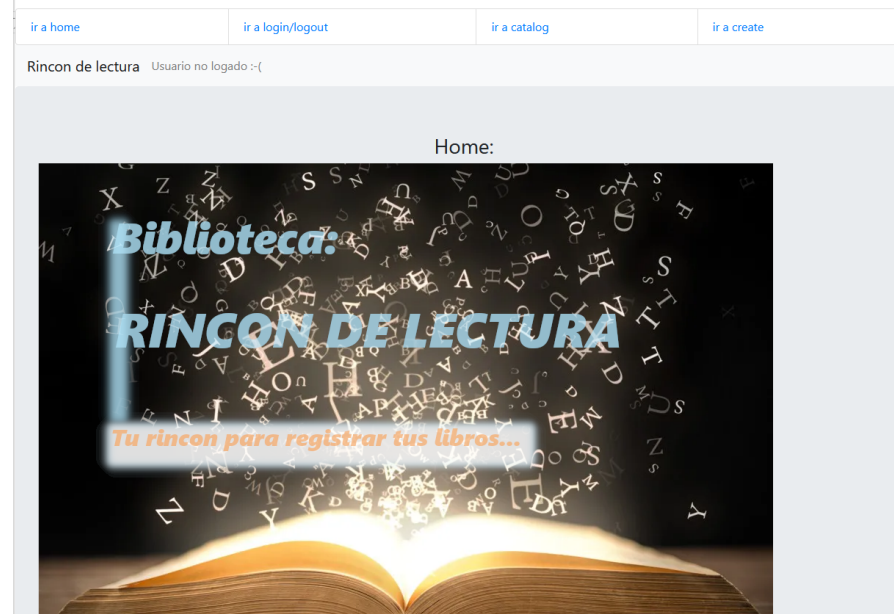
```
public function handle(Request $request, Closure $next)
{
    echo "Php -> Fecha: " . date("d") . " del " . date("m") . " de " . date("Y") . " - Php -> Hora: " . date("H:i:s");
    //echo "<br>";
    echo " // Laravel helper -> Fecha: " . today() . " - Laravel helper-> Hora: " . now();
    echo "<br>";
    //echo '<h5>' . date("d/m/y") . '</h5>'; Carboon
    return $next($request);
}
```

4.- Crud + Validacion:

Se crean todos los formularios para crear un CRUD:

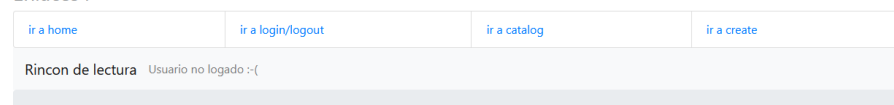
-Ya que no se interactúa con BBDD se generan los datos en un array que se utiliza fijo en la aplicación.

Enlaces :



Página de **Home**: En la parte superior se muestra el menú de navegación y el estado de login en un NavBar:

Enlaces :



Enlaces :

[ir a home](#)

[ir a login/logout](#)

[ir a catalog](#)

[ir a create](#)

Rincon de lectura

Usuario logado:

Catalog:

#	Título:	Autor:	Fecha:	Opinion:	Precio:	Show:	Edit:	Delete:
1	Reina roja	Juan Gómez-Jurado	2020-01-21	5	19	Show	Edit	Delete
2	Linea de fuego	Arturo Pérez-Reverte	2020-02-12	5	21	Show	Edit	Delete
3	Tierra	Eloy Moreno	2019-04-15	5	19	Show	Edit	Delete
4	Largo petalo de mar	Isabel Allende	2018-06-20	5	19	Show	Edit	Delete
5	Sidi	Arturo Pérez-Reverte	2017-09-15	5	19	Show	Edit	Delete

Pagina de **show/catalog**: de todos los registros. En esta página tenemos las opciones de **Show**: mostrar el detalle del registro, **Edit**: editar el registro, y **Delete**: para borrarlo.

Enlaces :

ir a home	ir a login/logout	ir a catalog	ir a create
Rincon de lectura Usuario no logado :-{			
Show:			
id: 1			
Título: Reina roja			
Autor: Juan Gómez-Jurado			
Fecha: 2021			
Opiniones: 5			
Precio: 19.85			

Pagina **show**: para mostrar el detalle especifico del registro seleccionado.

Enlaces :

ir a home	ir a login/logout	ir a catalog
---------------------------	-----------------------------------	------------------------------

Rincon de lectura Usuario logado:

Edit a book:

Ruta envio form por post: http://192.168.1.12/M1001/public/index.php/catalog/edit/1

ID: 1

Título: *(min 5, max 15)

Reina roja

Autor: *(min 5, max 15)

Juan Gómez-Jurado

Fecha: *(fecha valida)

21 / 01 / 2020

Opinion: *(num entre 1 -10)

5

Precio: *(numerico entre 1 - 200)

19

Los campos con * son obligatorios

Guardar

Pagina **edit abook**: para editar el registro seleccionado. El id no se puede cambiar. El resto de datos, se muestra el dato a modificar. También se realiza la validación. En caso de error de validación, se muestra un mensaje de advertencia del campo incorrecto, y se guardan los valores anteriores para que el usuario no tenga que introducirlos de nuevo.

Ruta envio form por post: http://192.168.1.1

ID: 1

Titulo: *(min 5, max 15)

*The titulo field is required.

Autor: *(min 5, max 15)

Juan Gómez-Jurado

Ejemplo de como se muestra el error de validacion de datos.

Enlaces :

ir a home

ir a login/logout

ir a catalog

ir a create

Rincon de lectura Usuario no logado :-{

Registro eliminado:

El registro con id: 3 ha sido eliminado...

[Regresar a home.](#)

En la opcion **delete** creamos un formulario con post, @csrf y @method('DELETE') y nos lleva a una vista que nos indica que registró ha sido eliminado correctamente.

Enlaces :

ir a home

ir a login/logout

ir a catalog

Rincon de lectura Usuario logado:

Create New Book:

Ruta envio form por post: http://192.168.1.12/M1001/public/index.php/catalog/create

Titulo: *(min 5, max 50)

Autor: *(min 5, max 50)

Fecha: *(fecha valida)

dd / mm / aaaa

Opinion: *(num entre 1 -10)

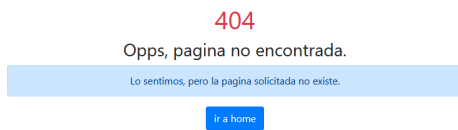
Precio: *(numerico entre 1 - 200)

Los campos con * son obligatorios

Guardar

Pagina **Create new book**: para crear un nuevo registro, que también realiza las validaciones.

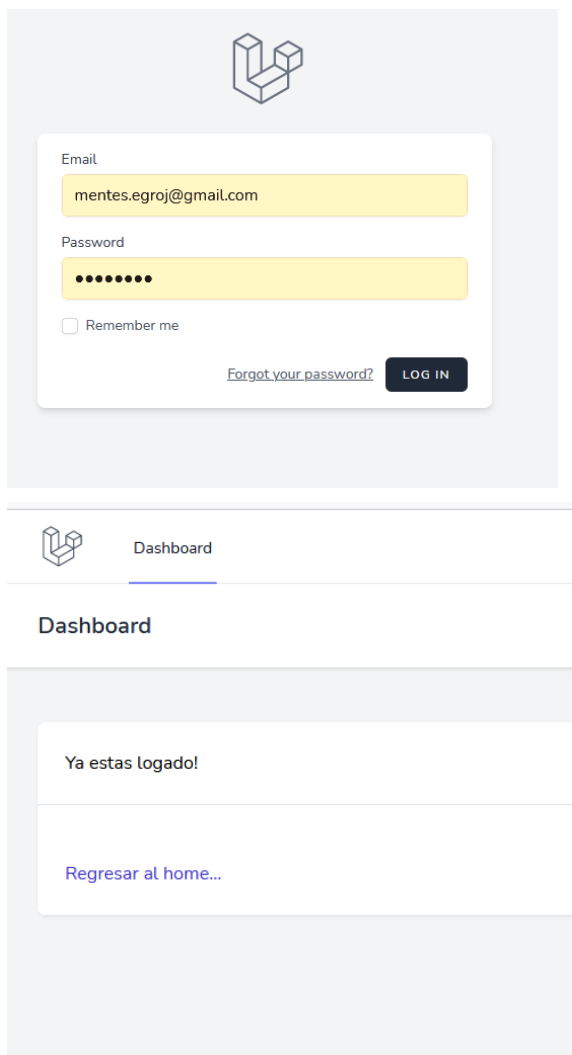
5.- Error 404 personalizado:



Se implementa la pagina de error personalizada, desde la que podemos regresar a home.

6.- Inicio sesion:

Se implementa el paquete **Auth**, que realiza todas estas tareas de **Login/logout/restauracion de password por email**.



Se realiza una modificacion de: `html/M1001/resources/views/dashboard.blade.php` para que aparezca el mensaje “Ya estas logado” y un link a la pagina de home.

7.- Cookies

Se añaden gestion de cookies para que en el navbar se muestre si el usuario esta logado,

ir a home	ir a login/logout
Rincon de lectura Usuario logado: MentEsgroj	

Mensaje de ejemplo del **navbar** cuando se está logueado.

ir a home	ir a login/logout
Rincon de lectura Usuario no logado :-(

Mensaje del **navbar** cuando no se está logado.