ESERCITAZIONE

AIUTINI / SOLUZIONI

AIUTINI O PER MEGLIO DIRE SOLUZIONI

1) Creare migrazione

php artisan make:migration create_articles_table
--create="articles"

Un messaggio ci dovrebbe avvisare della creazione della migrazione, verifichiamo che sia presente il file corrispondente nella cartella database/migrations.

AIUTINI O PER MEGLIO DIRE SOLUZIONI

2) Aggiungere i campi della tabella da migrare

Aprire la migrazione appena creata nella cartella database/migrations e dopo la riga

$table- >increments('id');

inserire

$table- >string('title');

$table- >text('body');

prima di $table- >timestamps();

AIUTINI O PER MEGLIO DIRE SOLUZIONI

3) Eseguire la migrazione
php artisan migrate

Veniamo informati con un messaggio che la migrazione ha avuto successo, altrimenti se abbiamo un errore PDO o SQL significa che non c'è connessione al Database:
1) abbiamo acceso la macchina virtuale con vagrant up?
2) abbiamo il giusto indirizzo e porta del database nel file .env?

AIUTINI O PER MEGLIO DIRE SOLUZIONI

4) Creare il model, l'entità corrispondente alla tabella
php artisan make:model Article

Verifichiamo che abbia creato correttamente la classe constatando la presenza del file Article.php nella cartella app.

AIUTINI O PER MEGLIO DIRE SOLUZIONI

5) Gestire la resouce nel router
Aggiungere al file routes.php nella cartella app/Http la seguente istruzione:
Route::resource('articles', 'ArticlesController');

Possiamo verificare che il router abbia correttamente preso la risorsa eseguendo da shell il seguente comando:
php artisan route:list

AIUTINI O PER MEGLIO DIRE SOLUZIONI

6) Creare il controller che gestisce le richieste
php artisan make:controller ArticlesController

Apriamo ora il file creato ArticlesController.php in app/Http/Controllers e copiamo il contenuto di tutti i metodi presenti in UsersController.php, avendo però ben cura di cambiare tutti i riferimenti a users in articles!

AIUTINI O PER MEGLIO DIRE SOLUZIONI

Aggiornare anche il file RouteServiceProvider che si trova nella cartella app/Providers prendendo spunto da Users aggiungendo

```
$router->bind('articles', function($id) {
    return \App\Article::where('id', $id)->firstOrFail();
});
```

Creare anche la Request specifica per Article, ArticleRequest

php artisan make:request ArticleRequest

AIUTINI O PER MEGLIO DIRE SOLUZIONI

Aprire il file UserRequest.php nella cartella app/Http/Requests

1) Nel metodo authorize(), ritornare il valore true al posto di false in modo che tutti siano autorizzati

2) Nel metodo rules(), definire le regole specifiche per articles prendendo spunto da UserRequest.php

AIUTINI O PER MEGLIO DIRE SOLUZIONI

7) Creare le view che visualizzano la pagine
Creare una copia della cartella users in resources/views rinominandola articles.
Modificare il contenuto delle view all'interno della cartella articles sostituendo i riferimenti da users ad articles.
Ricordarsi poi nel file form.blade.php di gestire correttamente i campi della maschera in base a title e body, eventualmente aggiustando anche l'import di form in create.blade.php e edit.

AIUTINI O PER MEGLIO DIRE SOLUZIONI

8) Creare un seeder usando la factory
php artisan make:seeder ArticlesSeeder

Modificare il file DatabaseSeeder.php in database/seeds aggiungendo questo comando prima di Model::reguard():
$this- >call(ArticlesSeeder::class);
ed eventualmente, se vogliamo cancellare prima di inserire:
DB::table('articles')- >delete();

AIUTINI O PER MEGLIO DIRE SOLUZIONI

Modificare il file ModelFactory.php in database/factories aggiungendo alla fine:

```
$factory- >define(App\Article::class, function (Faker\Generator $faker) {
  return [
      'title' => $faker- >sentence(6),
      'body' => $faker- >paragraph(9)
  ];});
```

AIUTINI O PER MEGLIO DIRE SOLUZIONI

Modificare il file ArticlesSeeder.php in database/seeds aggiungendo all'interno del metodo run() il seguente codice:
factory(App\Article::class, 50)- >create();

Per eseguire il seed ed inserire (per esempio) 50 articoli eseguire sempre a riga di comando:
php artisan db:seed
Veniamo avvisati di quali seed vengono effettivamente eseguiti.

GRAZIE

per l'attenzione!