

1. Atsidaryti su VC susikurtą projekto folderį.
2. phpMyAdmin susikurti naują duomenų bazę. Koduotė utf8mb4_bin.
3. Atsidaryt .env failą ir pakeisti.
DB_DATABASE=vardas
DB_USERNAME=root
DB_PASSWORD=mysql (mac) (arba palikti tuščią windows)

4. Terminale:
composer require laravel/ui
php artisan ui vue --auth
npm install
npm run dev
Visur LAUKTI po 5-7 min!

php artisan make:model Book --all (pavadinimas pagal uždavinio sąlygą)
php artisan make:model Author --all (pavadinimas pagal uždavinio sąlygą)

Rašyti **vienaskaita!**

5. Atsidaryti *database/migration* ir susirasti book ir author migracijų failus
6. Pildom up funkciją:
varchar keičiam į string;
visų tipų integer (pvz tinyint) keičiam tiksliai į integer ir nerašom jo dydžio;
visus stulpelius _id galūne darome unsignedBigInteger tipo ir nerašom jo dydžio;
7. Terminale: php artisan migrate:fresh

8. Į migracijos failą idedame:
\$table->foreign('author_id')->references('id')->on('authors');
9. Migruojam. Jeigu gauname klaidą, pervadinam failus taip, kad failas BE author_id būtų pirmesnis.
10. Atsidarome routes/web.php

Kopijuojam kodą į failo apačią, pagal sąlyga pakeisdami Author ir author:

```
Route::group(['prefix' => 'authors'], function() {  
    Route::get('', 'AuthorController@index')->name('author.index');  
    Route::get('create', 'AuthorController@create')->name('author.create');  
    Route::post('store', 'AuthorController@store')->name('author.store');  
    Route::get('edit/{author}', 'AuthorController@edit')->name('author.edit');  
    Route::post('update/{author}', 'AuthorController@update')->name('author.update');  
    Route::post('delete/{author}', 'AuthorController@destroy')->name('author.destroy');  
    Route::get('show/{author}', 'AuthorController@show')->name('author.show');  
});
```

11. Atsidarom *app/Http/Controllers* failą AuthorController.php (tą failą, kurio migracijoje nėra author_id)
12. Pradedame nuo create() kodas:

```
return view('author.create');
```

13. Atsidarome *resources/views* folderį ir ten sukuriame folderį *author* o tame folderyje sukuriame tris tuščius failus:

create.blade.php

edit.blade.php

index.blade.php

14. *create.blade.php* faile darome html formą:

```
<form method="POST" action="{{route('author.store')}}">
    Name: <input type="text" name="author_name">
    Surname: <input type="text" name="author_surname">
    @csrf
    <button type="submit">ADD</button>
</form>
```

15. Tikriname veikimą adresu: <http://localhost/lib/public/authors/create> (čia /lib/ jūsų laravel folderio pavadinimas)

16. Atsidarom *app/Http/Controllers* failą *AuthorController.php* metodas *store()* kodas:

```
$author = new Author;
$author->name = $request->author_name;
$author->surname = $request->author_surname;
$author->save();
return redirect()->route('author.index');
```

17. Įrašome author vardą ir pavardę, spaudžiam, gauname tuščią langą.

18. Atsidarom *app/Http/Controllers* failą *AuthorController.php* metodas *index()* kodas:

```
$authors = Author::all();
return view('author.index', ['authors' => $authors]);
```

19. *resources\views\author\index.blade.php* faile dedame html:

```
@foreach ($authors as $author)
    {{ $author->name }} {{ $author->surname }}<br>
@endforeach
```

20. Tikriname veikimą adresu: <http://localhost/lib/public/authors> (čia /lib/ jūsų laravel folderio pavadinimas) Turi rodyti įrašytus autorius

21. *resources\views\author\edit.blade.php* faile dedame html:

```
<form method="POST" action="{{route('author.update', [$author])}}">
    Name: <input type="text" name="author_name" value="{{ $author->name }}">
    Surname: <input type="text" name="author_surname" value="{{ $author->surname }}">
    @csrf
    <button type="submit">EDIT</button>
</form>
```

22. Atsidarom *app/Http/Controllers* failą *AuthorController.php* metodas *edit()* kodas:

```
return view('author.edit', ['author' => $author]);
```

23. *resources\views\author\index.blade.php* faile redaguojame html:

```
@foreach ($authors as $author)
    <a href="{{route('author.edit', [$author])}}">{{ $author->name }}
    {{ $author->surname }}</a><br>
@endforeach
```

24. Spaudžiam ant linko, turi atsidaryti užpildyta forma

25. Atsidarom *app\Http\Controllers* failą *AuthorController.php* metodas *update()* kodas:

```
$author->name = $request->author_name;
$author->surname = $request->author_surname;
$author->save();
return redirect()->route('author.index');
```

26. Tikrinam veikimą. Turi vardai atsinaujinti.

27. *resources\views\author\index.blade.php* faile vėl redaguojame html:

```
@foreach ($authors as $author)
    <a href="{{route('author.edit', [$author])}}">{{ $author->name }}
    {{ $author->surname }}</a>
    <form method="POST" action="{{route('author.destroy', [$author])}}">
        @csrf
        <button type="submit">DELETE</button>
    </form>
    <br>
@endforeach
```

28. Atsidarom *app\Http\Controllers* failą *AuthorController.php* metodas *destroy()* kodas:

```
$author->delete();
return redirect()->route('author.index');
```

29. Tikriname trynimo veikimą.

30. Pradedame books lentelę.

31. Atsidarome *routes\web.php*

Kopijuojam kodą į failo apačią:

```
Route::group(['prefix' => 'books'], function() {
    Route::get('', 'BookController@index')->name('book.index');
    Route::get('create', 'BookController@create')->name('book.create');
    Route::post('store', 'BookController@store')->name('book.store');
    Route::get('edit/{book}', 'BookController@edit')->name('book.edit');
    Route::post('update/{book}', 'BookController@update')->name('book.update');
    Route::post('/delete/{book}', 'BookController@destroy')->name('book.destroy');
    Route::get('show/{book}', 'BookController@show')->name('book.show');
});
```

32. Atsidarom *app\Http\Controllers* failą *BookController.php* (tą failą, kurio migracijos faile yra id)

33. Pradedame nuo *create()* kodas:

```
return view('book.create');
```

34. Atsidarome *resources\views* folderį ir ten sukuriame folderį *book* į tą folderį perkopijuojame failus iš *author* folderio:

create.blade.php

edit.blade.php

index.blade.php

35. *create.blade.php* faile darome html formą:

```
<form method="POST" action="{{route('book.store')}}">
    Title: <input type="text" name="book_title">
    ISBN: <input type="text" name="book_isbn">
    Pages: <input type="text" name="book_pages">
    About: <textarea name="book_about"></textarea>
    <select name="author_id">
        @foreach ($authors as $author)
            <option value="{{ $author->id }}">{{ $author->name }}
            {{ $author->surname }}</option>
        @endforeach
    </select>
    @csrf
    <button type="submit">ADD</button>
</form>
```

36. Atsidarom *app\Http\Controllers* failą *BookController.php* ir papildome *create()* metoda:

```
$authors = Author::all();
return view('book.create', ['authors' => $authors]);
```

37. Tame pačiame faile viršuje pridedame:

```
use App\Author;
```

38. Tikriname veikimą adresu: <http://localhost/lib/public/books/create> (čia /lib/ jūsų laravel folderio pavadinimas)

39. Atsidarom *app\Http\Controllers* failą *BookController.php* metodas *store()* kodas:

```
$book = new Book;
$book->title = $request->book_title;
$book->isbn = $request->book_isbn;
$book->pages = $request->book_pages;
$book->about = $request->book_about;
$book->author_id = $request->author_id;
$book->save();
return redirect()->route('book.index');
```

40. Užpildome formą, spaudžiam, gaunam tuščią langą.

41. Atsidarom *app\Http\Controllers* failą *BookController.php* metodas *index()* kodas:

```
$books = Book::all();
return view('book.index', ['books' => $books]);
```

42. *resources\views\book\index.blade.php* faile dedame html:

```
@foreach ($books as $book)
    <a href="{{route('book.edit',[$book])}}">{{ $book->title }}</a>
    <form method="POST" action="{{route('book.destroy', [$book])}}">
        @csrf
        <button type="submit">DELETE</button>
```

```

</form>
<br>
@endforeach

```

43. Tikriname veikimą adresu: <http://localhost/lib/public/books> (čia /lib/ jūsų laravel folderio pavadinimas) Turi rodyti įrašytas knygas

44. Reikia padaryti kad rodytų knygos autorių.

45. Atsidarom *app\Book.php* ir tame faile sukuriame naują funkciją:

```

public function bookAuthor()
{
    return $this->belongsTo('App\Author', 'author_id', 'id');
}

```

46. *resources\views\book\index.blade.php* faile padarome papildymą:

```

@foreach ($books as $book)
    <a href="{{route('book.edit', [$book])}}">{{ $book->title }} {{ $book->bookAuthor->name }}
    {{ $book->bookAuthor->surname }}</a>
    <form method="POST" action="{{route('book.destroy', [$book])}}">
        @csrf
        <button type="submit">DELETE</button>
    </form>
    <br>
@endforeach

```

47. *resources\views\book\edit.blade.php* faile dedame html:

```

<form method="POST" action="{{route('book.update', [$book])}}">
    Title: <input type="text" name="book_title" value="{{ $book->title }}">
    ISBN: <input type="text" name="book_isbn" value="{{ $book->isbn }}">
    Pages: <input type="text" name="book_pages" value="{{ $book->pages }}">
    About: <textarea name="book_about">{{ $book->about }}</textarea>
    <select name="author_id">
        @foreach ($authors as $author)
            <option value="{{ $author->id }}" @if($author->id == $book->author_id)
selected @endif>
                {{ $author->name }} {{ $author->surname }}
            </option>
        @endforeach
    </select>
    @csrf
    <button type="submit">EDIT</button>
</form>

```

48. Atsidarom *app\Http\Controllers* failą *BookController.php* metodas *edit()* kodas:

```

$authors = Author::all();
return view('book.edit', ['book' => $book, 'authors' => $authors]);

```

49. Spaudžiam ant linko, turi atsidaryti užpildyta forma

50. Atsidarom *app\Http\Controllers* failą *BookController.php* metodas *update()* kodas:

```

$book->title = $request->book_title;

```

```

$book->isbn = $request->book_isbn;
$book->pages = $request->book_pages;
$book->about = $request->book_about;
$book->author_id = $request->author_id;
$book->save();
return redirect()->route('book.index');

```

51. Tikrinam veikimą. Turi vardai atsinaujinti.

52. Atsidarom *app\Http\Controllers* failą *BookController.php* metodus *destroy()* kodas:

```

$book->delete();
return redirect()->route('book.index');

```

53. Tikriname trynimo veikimą.

54. Bookai veikia!

55. Papildome *app\Http\Controllers* failą *AuthorController.php* metoda *destroy()*:

```

if($author->authorBooks->count()){
    return 'Trinti negalima, nes turi knygu';
}
$author->delete();
return redirect()->route('author.index');

```

56. Atsidarom *app\Author.php* ir tame faile sukuriame naują funkciją:

```

public function authorBooks()
{
    return $this->hasMany('App\Book', 'author_id', 'id');
}

```

57. Tikrinam trynimą.

58. Viskas.