

Larasocial – közösségi oldal

Webes ismeretek - Beadandó feladat

Tartalom

Általános leírás	2
Models	2
User model	2
Post model	2
Migrations	2
Controlleres	2
Routes	3
Views	3
Javasolt feladatlépés sorrend	3

Általános leírás

A kész projekt elérhető itt, ezt kellene megvalósítanotok a beadandóban is:

<https://larasocial.composite-solutions.eu/>

A projekt egy közösségi oldal lesz, gyakorlatilag egy minimál design facebookot fogok kérni tőletek.

Fontos dolgok az oldalon:

- **Authentikáció** - regisztráció nélkül nem érhető el semmi;
- Bejelentkezés után az oldal a **kezdőlapra** irányítson, ahol a **poszt feltöltésének ideje** alapján **csökkenő** sorrendben találhatók a különböző emberek posztjai;
- A jobb oldalon láthatók a **regisztrált felhasználók**, a saját nevünk értelemszerűen nem szerepel ebben a listában. A nevekre kattintva a felhasználók **profiljai** találhatók a saját **bejegyzéseikkel**, szintén időrendben **csökkenő** sorban;
- A jobb felső sarokban a nevünkre kattintva lehetőségünk van egy legördülő menüből kiválasztani, hogy **kijelentkezni** szeretnénk, vagy a **profilunkat** szeretnénk megtekinteni;
- A saját profil oldalunkon lehetőségünk van **kitörölni** a saját magunk által feltöltött cikkeket;
- A posztokat lehetőségünk van **like**-olni, itt a zárójelben a szám mutatja, hogy hányszor lett a gombra rákattintva

Models

A projektben összesen 2 db model van, ezek a User és a Post modellek.

User model

A kézzel kialakítandó metódus a User – Post kapcsolat, ami egy one-to-many kapcsolat, vagyis egy felhasználónak lehet több posztja, viszont egy poszt nem lehet több felhasználóé (**hasMany**).

Post model

A kézzel kialakítandó metódus a Post - user kapcsolat, ami egy many-to-one kapcsolat, vagyis egy felhasználónak lehet több posztja, viszont egy poszt nem lehet több felhasználóé (**belongsTo**). A User model-ben kialakított kapcsolat a másik irányból nézve.

Migrations

A Users migration már eleve kész van, ezzel nem kell foglalkoznunk. Szükséges viszont létrehozni a **Post model létrehozásakor** egy migration-t is hozzá (**php artisan make:model Post -mcr**), ebben pedig kell egy **kapcsolómező** a user felé az ehhez tartozó **idegen kulccsal**, egy szöveges rész a bejegyzésnek és egy szám változó a like-oknak.

Az időbélyeg automatikusan hozzáíródik, így ezzel nem kell foglalkozni.

Controlleres

Három controllerre lesz szükségünk, és egy picit kell majd módosítanunk (az oldal ne view-t töltsön be, hanem redirecteljen a kezdőlapra) a negyediket, ami létrejön az autentikáció létrehozásakor (HomeController). Ezek a következők:

- PageController – egy **index** függvényre lesz szükség benne összesen, ami a **kezdőlapot** betölti és ráírja a postokat.!!! Hint: `$posts = Post::orderBy('created_at', 'DESC')->get();` !!!

- PostController – **három** metódust kell ebben implementálni, egyet ami ment az adatbázisba (**store**), egyet ami töröl belőle (**destroy**), és egyet ami a **like**-ok hozzáadásáért felel a gomb megnyomásakor
- UserController – egy **show** függvényre lesz szükségünk, ami betölti a kapott **id** alapján az adott felhasználó **profilját**.

Routes

A routes/web.php-ban a következő sorok kelleni fognak, ezt segítségképpen megadom:

```
// Kezdőlap
Route::get('/', 'PageController@index')->name('pages.index')->middleware('auth');

// Profil oldal
Route::get('/user/{id}', 'UserController@show')->name('users.show')->middleware('auth');

// Posztokhoz tartozó route-ok
Route::post('/posts', 'PostController@store')->name('posts.store')->middleware('auth');
Route::delete('/posts/{id}', 'PostController@destroy')->name('posts.delete')->middleware('auth');
Route::get('/posts/{id}/like', 'PostController@like')->name('posts.like')->middleware('auth');
```

Views

A következő fileszerkezetet kell elérnünk:

- views
 - auth **Nem piszkáljuk**
 - inc
 - messages.blade.php **A redirect üzenetek itt jelennek meg**
 - users.blade.php **A jobb oldali user lista, ami include-olható később**
 - layouts
 - app.blade.php **Layout file-unk, amit örököl minden oldal**
 - pages
 - index.blade.php **Kezdőlap, ezen a poszt megosztásának lehetősége és az összes poszt listázva**
 - users
 - show.blade.php **Felhasználó profil oldala**

Az elrendezéshez nem ragaszkodom, csak mintaképpen készítettem el az oldalt, viszont a funkcionalitáshoz igen.

Javasolt feladatlépés sorrend

(én így csináltam, lehet máshogy is ha kényelmesebb)

1. Laravel 7 install **composer** paranccsal
2. **adatbázis** létrehozása, **.env** file kitöltése
3. Authentikáció install (**dokumentáció**: <https://laravel.com/docs/7.x/authentication>)
4. **Modellek** és **Controllerek** létrehozása
 - a. model létrehozás **-mcr** kapcsolóval, létrejön a **migration** is és a **controller** is
 - b. **migration** kitöltése értelemszerűen a fent leírtaknak megfelelően
 - c. **php artisan migrate**

- d. **Controller** metódusok implementálása
- 5. View-k létrehozása és controllerekkel a kapcsolat kialakítása (GET, POST, PUT, DELETE requestekkel és formokkal)

Egyedi megoldások és reszponzív megjelenés plusz pontot jelentenek.

A böngészőben a minta oldalon lévő kódok mind a sztenderd bootstrap elemekből állnak, nem szükséges css-t írni hozzá. Az oldalon az F12 billentyűvel előhívható a böngésző fejlesztői eszköze, itt sok hasznos információ látható, illetve bizonyos frontend kódrészletek is lophatók tőlem (vagy bárki mástól...).

Várom a feladatokat, elakadás esetén e-mail-ben elérhető vagyok! 😊