1. **laravel projeyi olusturuyoruz**

- laravel new blog\_oguz\_v1

1. **Projeye vebsite dosyasini dahil ediyoruz**

- blog\_oguz\_v1->public-> dosyasi icerisinde front isminde dosya acip, icerisine template vebsitesinin butun dosyalarini ekliyoruz

1. **View kisminda front islemlerinin rotalari icin hazirlik**

- resources->views->welcome.blade.php isminde default gelen dosyayi siliyoruz

- resources->views-> icerisine front isminde klasor olusturuyoruz

- resources->views->front icerisine layouts isminde klasor olusturuyoruz

- resources->views->front-> icerisine homepage.blade.php isminde dosya olusturuyoruz

1. **İndex.php parcalama**

- public->front->index.html -dosyasini kopyalayip views icerisinde yaratdigimiz homepage.blade.php dosyasi icerisine atiyoruz

- resources->views->front->layouts icersinde header.blade.php, footer.blade.php, master.blade.php isminde dosya yaratiyoruz

- header.blade.php dosyasinda homepage icersinde header kismina ait olanlari alip atiyoruz

- footer.blade.php dosyasinda homepage icersinde footer kismina ait olanlari alip atiyoruz

-master.blade.php dosyasinda sayfa tarasimi yapiyoruz. (Daha detayli analmak icin dosyaya bak)

1. **İlk dosya calistirma**

- routes->web.php – ile butun proje zamani dosyalarin route-larini belirliyoruz.

- route dediyimiz kisim sayfamizin yukari link boslugu kisminda yazilan bolumde ne yazilirsa, ona uygun hangi sayfa acilacagi belirtilir.

- routes->web.php dosyamizda anasayfa acildigi zaman bu “/” – bos slash anlamina gelir, view klasoru icerisinde bulunan front->homepage.blade.php dosyasini cagiriyoruz. (Daha detayli anlamak icin dosyaya bak)

1. **Css,js dosya yolu ayarlama**

- resources->views->front->layouts->header.blade.php ve resources->views->front->layouts->footer.blade.php dosyalarini aciyoruz

- css,js dosya eklentilerine bakiyoruz, cdn link ile eklenmeyenlerde duzeltmeler yapiyoruz {{asset}} -comutu ile (Daha detayli anlamak icin dosyaya bak)

1. **<title></title> - alanini her sayfaya ozel yapma**

- views->front->layouts->header.blade.php – dosyasina geliyoruz

- <title></title> alani icerisine @yield(‘title’) komutu ile kullanima hazirliyoruz

1. **Veritabani baglama**

- phpmyadmin(ve s.) database olusturuyoruz

- proje icerisinde bulunan .env dosyasina gidiyoruz, orada database host,name, root,password dahil ediyoruz

1. **Veritablosu olusturma**

- veri tablosu olusturmak icin once migration olusturuyoruz

- migration ismi ile tablo ismi ayni olur genelde

- migration-lar tabloda olacak sutunlar, hangi veri tipinde ve basqa ozellikleri belirtilmek icin create olunur

- (Daha detayli anlamak icin dosyaya bak)

- terminalde migration olusturma comutu asagidaki gibidir

php artisan make:migration migration\_name --create=table\_name

- migration icerisine sutun isimlerini ve ozelliklerini ekledikden sonra terminalde asagidaki komut calistirilir.

php artisan migrate

- bu zaman database-de tablo ve icerisinde sutunlari olusur.

1. **Seeder kullanimi**

- daha sonra terminal ile seeder olusturuyoruz

php artisan make:seeder NameSeeder

- tabloda olan sutunlara veriler eklemek icin seeder olusturulur

- olusan seeder dosyasi database->seeder klasoru icerisinde yer alir

- seeder icerisinde en uste use Illuminate\Support\Facades\DB; yazilir.

- seeder icerisinde sutun ismi ve alicagi deger yazilir.

- (Daha detayli anlamak icin dosyaya bak)

- daha sonra olusturdugumuz bu seeder-i butun seeder-ler cagirilan DatabaseSeeder.php dosyasi icerisine ekliyoruz ve asagidaki kodu run() fonksiyon icerisine ekliyoruz

$this->call(NameSeeder::class);

- daha sonra terminalde asagidaki kodu ekliyoruz

php artisan db:seed

1. **Onemli NOT:**

- Tablolari ve iceriyini tamamen silip yeniden ekelmek isyoruzsa terminalde bu komutu calisdiriyoruz. Bu durumda hem tablolar yeniden olusur, hemde seed calisarak stunlara verileri yeniden ekler

php artisan migrate:refresh --seed

1. **Controller olusturuyoruz**

- front ve back bir-birine karismasin diye controller olustururken front dosyasi ve back dosyasi yolunu gostererek olusturacagiz

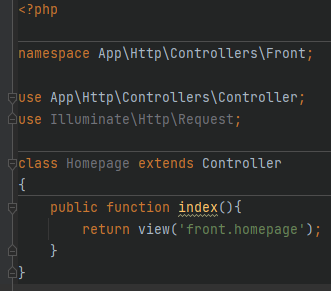
- controller olusturmak icin terminalde bu komutu yaziyoruz

php artisan make:controller Front/Homepage

- Homepage controlleri bu sayfaya ait fonksiyonlari yonetmek icin kullanicaz

- Homepage controlleri web.php dosyasinda rotalari eklemek icin kullanicaz

- Homepage controlleri iceirisne ilk index() function ekliyoruz. Bu fonksiyonla return view komutu ile anasayfayi cagiriyoruz. Daha oncesinde bu fonksiyonu web.php dosyasinda route kisminda yazmistik. Fonksiyonlarin hepsini kontroller icrisinde birlestiriyoruz



- web.php dosyasinda ilk basda yazdigimiz route islemini controllerden yoneteciyimiz icin orada yazacagimiz kodlar asagidaki gibidir



1. **Veritabani icin model olusturma**

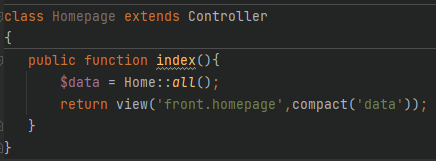
- Veri tabanindaki tablo verilerini ala bilmek icin model olusturuyoruz. Terminalde asagidaki komutu dahil ediyoruz

php artisan make:model ModelName

- Modeli kullana bilmek icin yaratdigimiz Controller icerisinde en uste

use App\Models\ModelName; kodunu ekleyerek tanimliyoruz

- Veri tabanindaki butun datalari almak icin tanimladigimiz fonksiyon icerisinde $data = ModelName::all() kodu kullaniyoruz, daha sonra compact ile aldigimiz verileri kullanilir hale getiriyoruz. Asagidaki orneye bakin



$ php artisan make:migration add\_column\_to\_portfolios

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd

1. **laravel projeyi olusturuyoruz**

- dfd