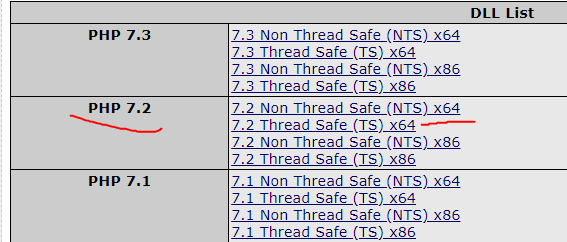
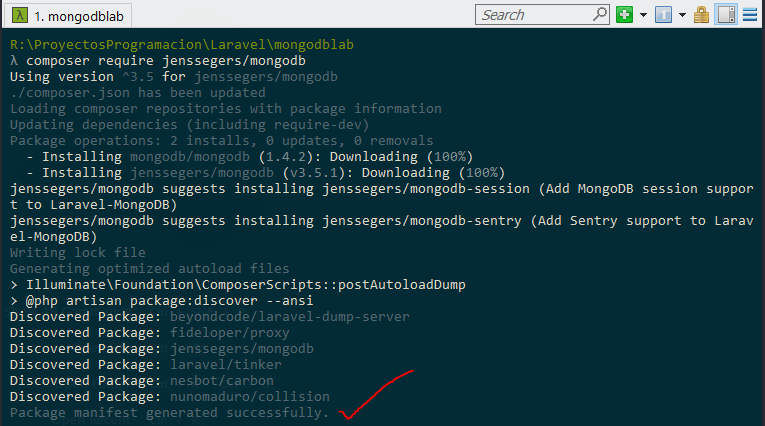
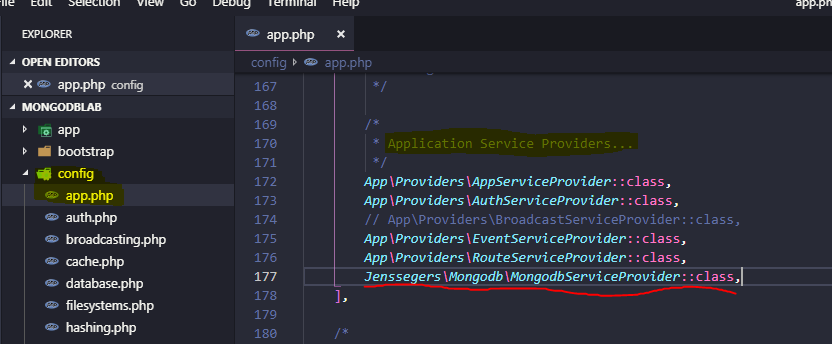
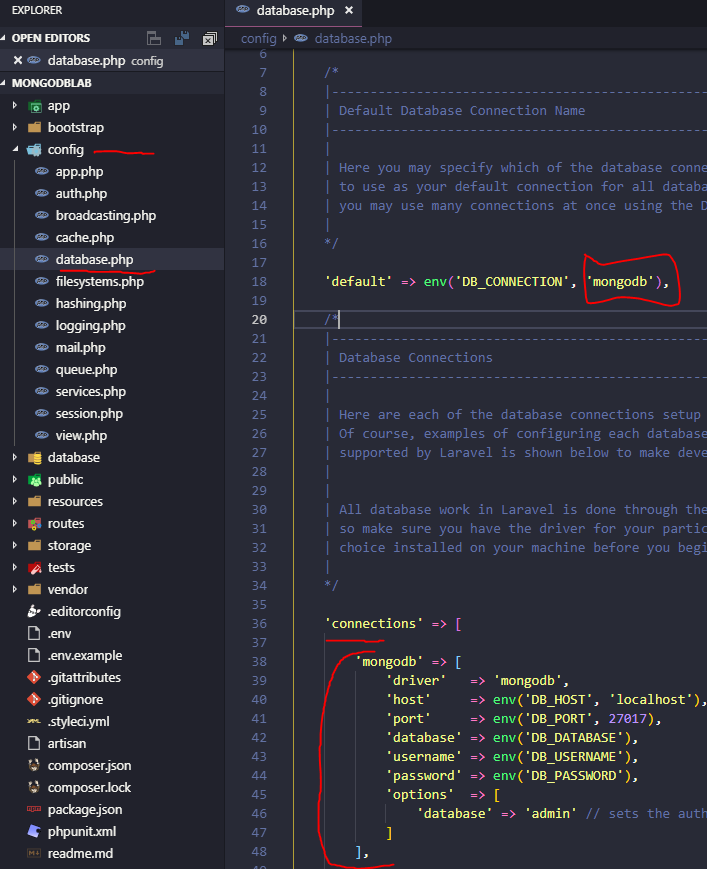
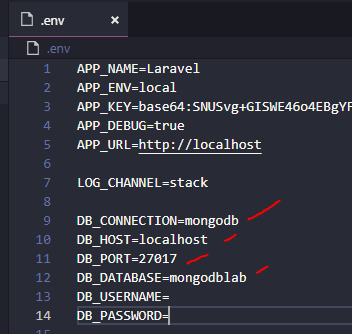
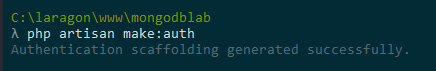
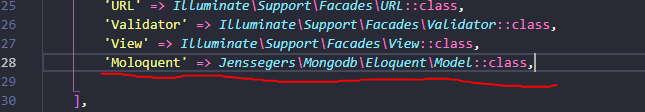
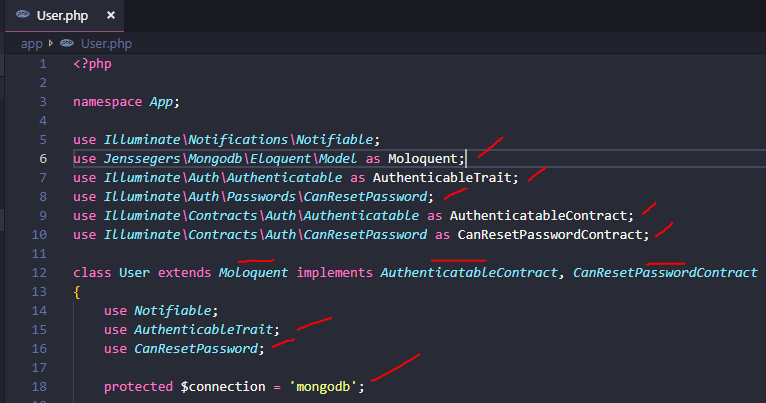
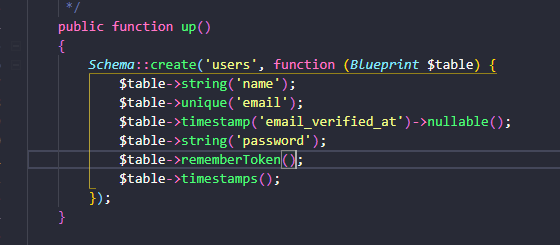
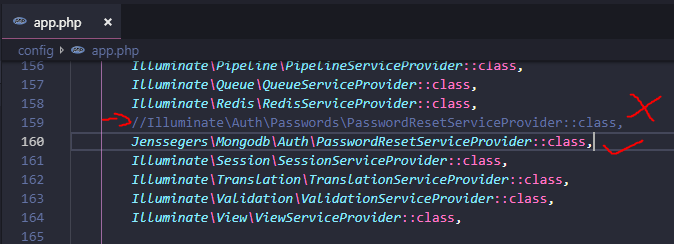
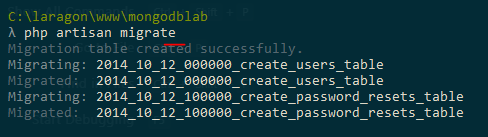
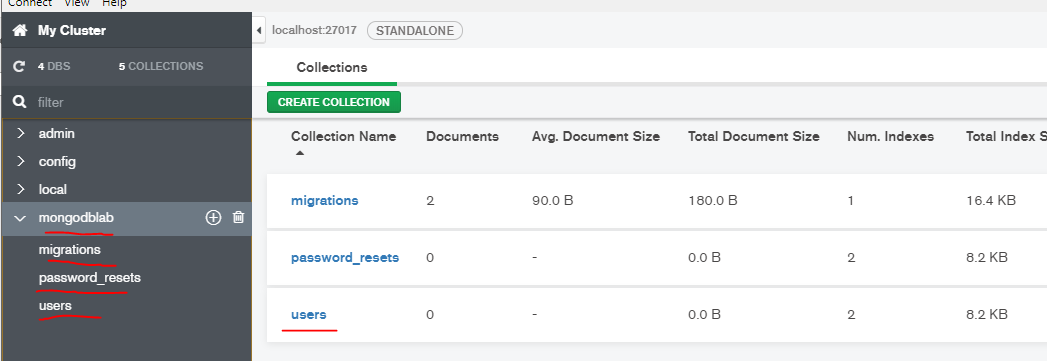
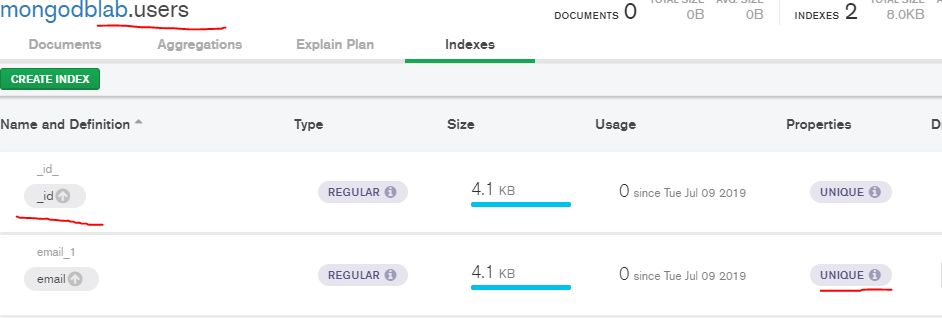
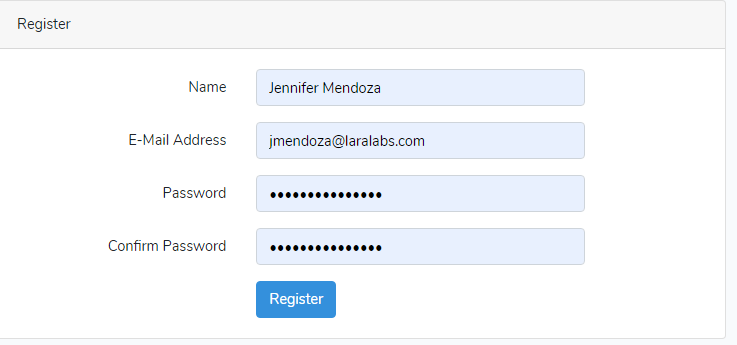
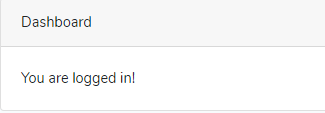
Laravel con MongoDB

1. Creación del proyecto en laravel con composer:  
    > composer create-project --prefer-dist laravel/laravel blog
2. Descargamos la versión mas reciente de mongodb.dll para php.ini:  
   > <https://pecl.php.net/package/mongodb>  
   En el caso de laragon, tiene que ir en la ruta de php, y tiene que ser la librería de x64 bits.  
     
   Laragon usa 7.2 y sus librerias por 64 bits.
3. Instalacion de Moloquent para trabajar con modelos en laravel y mongo:  
    > composer require jenssegers/mongodb  
     
   
4. Abrimos el proyecto de laravel con cualquier editor y colocamos en config/app.php el siguiente proveedor de servicios: **Jenssegers\Mongodb\MongodbServiceProvider::class**,   
     
   
5. Agregamos las propiedades de conexión a config/database.php:  
     
   
6. Ahora configuramos las propiedades de conexión del archivo .env:  
     
   
7. Creamos la autentificación básica por defecto de laravel para probar:  
   
8. Agregamos un alias para extender de eloquent usando la palabra Moloquent y de este modo entender que es un modelo de mongoDB, abrimos config/app.php y en el apartado de aliases agregamos lo siguiente:  
     
     
   Esto viene referido en la documentación de laravel-mongodb de github.
9. Modificamos el modelo User.php de Auth, para adaptarlo a un modelo de eloquent con mongodb:  
     
   
10. Modificamos la función UP de la migración de users, quedando de la siguiente manera:  
      
      
      
    La documentación de laravel-mongodb nos dice que por defecto cada tabla debe tener un campo id de tipo index, por lo cual, se omite aquí su agregación.
11. Modificamos la clase que nos provee del reinicio de contraseñas por defecto y especificamos la de laravel-mongodb, editamos el archivo config/app.php dejándolo de la siguiente manera:  
      
    
12. Ejecutamos las migraciones para comprobar que se genere bien la colección con los datos:  
      
    
13. Comprobamos en Compass si se ejecutó correctamente:  
      
      
      
    Las migraciones se ejecutaron satisfactoriamente, esto quiere decir que ya podemos trabajar con mongodb en nuestro proyecto de laravel.
14. En nuestra tabla users, efectivamente tenemos el campo id de tipo index por defecto, y el correo unique:  
      
    
15. Registramos un usuario desde nuestro sitio en laravel:  
      
      
      
    Si se registra bien, nos iniciara sesión automáticamente:  
      
      
    Y en nuestra tabla users, se verá reflejado el usuario:  
    