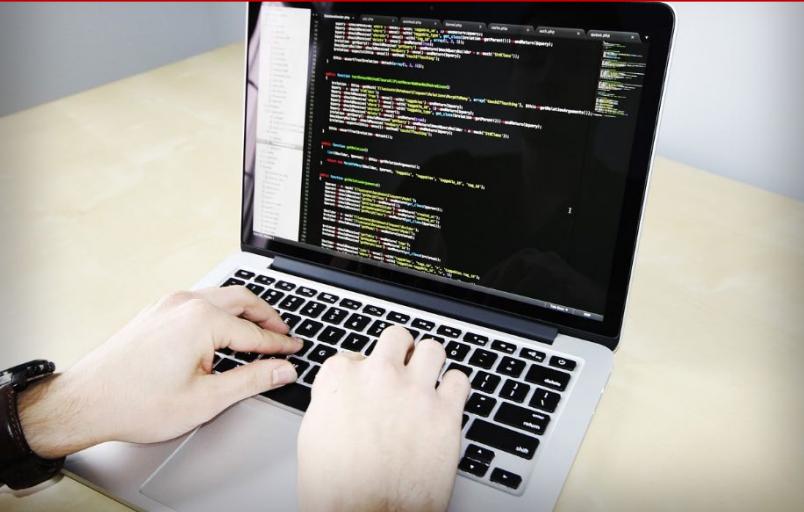


Laravel



```
REACT_MERGE_MODULE_CONTEXTS
REACT_PORTAL_TYPE
REACT_PROFILER_TYPE
REACT_PROVIDER_TYPE
REACT_STRICT_MODE_TYPE
REACT_SUSPENSE_TYPE
} from 'shared/ReactSymbols';
import {isValidElementType} from 'shared/elementTypeUtil';
if (typeof object === 'object' && object !== null) {
  const $typeof = object.$typeof;
  switch ($typeof) {
    case REACT_ELEMENT_TYPE:
      const type = object.type;
      switch (type) {
        case REACT_FRAGMENT_TYPE:
        case REACT_PROFILER_TYPE:
        case REACT_STRICT_MODE_TYPE:
        case REACT_SUSPENSE_TYPE:
          return type;
        default:
          const $typeofType = type.$typeof;
          switch ($typeofType) {
            case REACT_CONTEXT_TYPE:
            case REACT_FORWARD_REF_TYPE:
            case REACT_LAZY_TYPE:
            case REACT_MEMO_TYPE:
```

Arbeiten mit CMD

cls -> Eingabefeld löschen

```
Eingabeaufforderung

C:\Users\User>C:\xampp\php\php.exe -v
PHP 7.4.16 (cli) (built: Mar 2 2021 14:06:15) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\User>C:\xampp\php\php.exe -v
PHP 7.4.16 (cli) (built: Mar 2 2021 14:06:15) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\User>
C:\Users\User>C:\xampp\php\php.exe -v
PHP 7.4.16 (cli) (built: Mar 2 2021 14:06:15) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\User>C:\xampp\php\php.exe -v
PHP 7.4.16 (cli) (built: Mar 2 2021 14:06:15) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

C:\Users\User>
C:\Users\User>C:\xampp\php\php.exe -v
PHP 7.4.16 (cli) (built: Mar 2 2021 14:06:15) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies

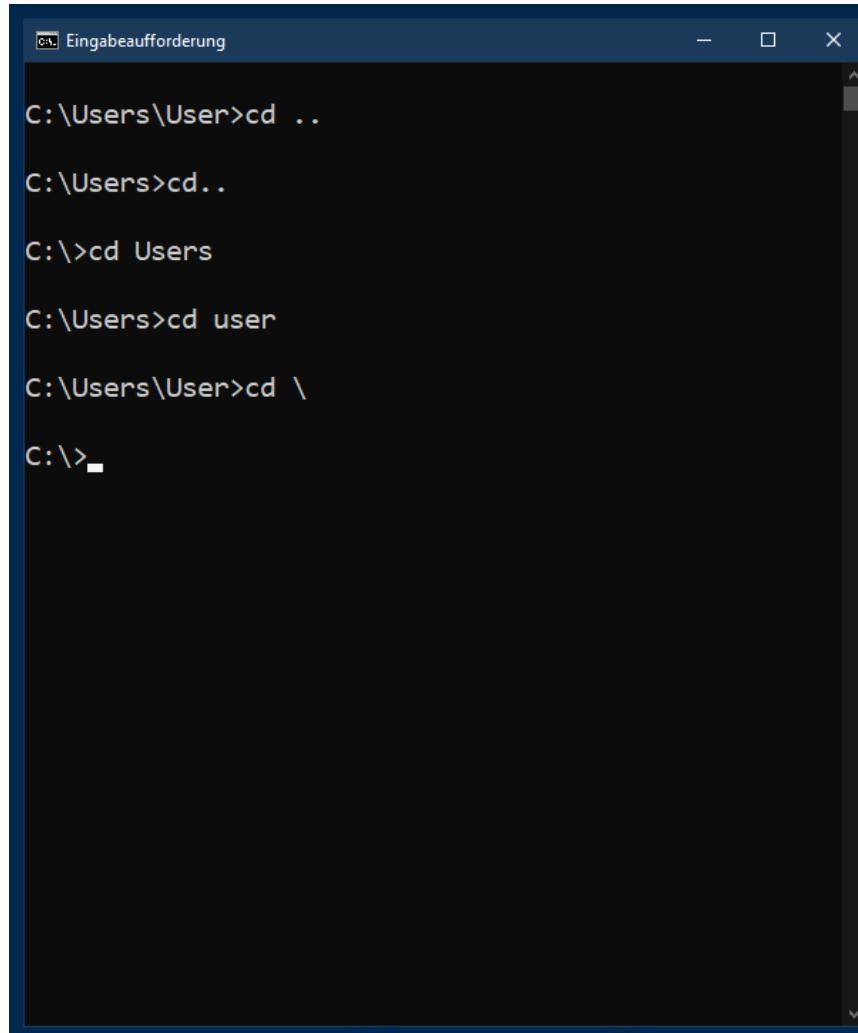
C:\Users\User>
```

```
Eingabeaufforderung

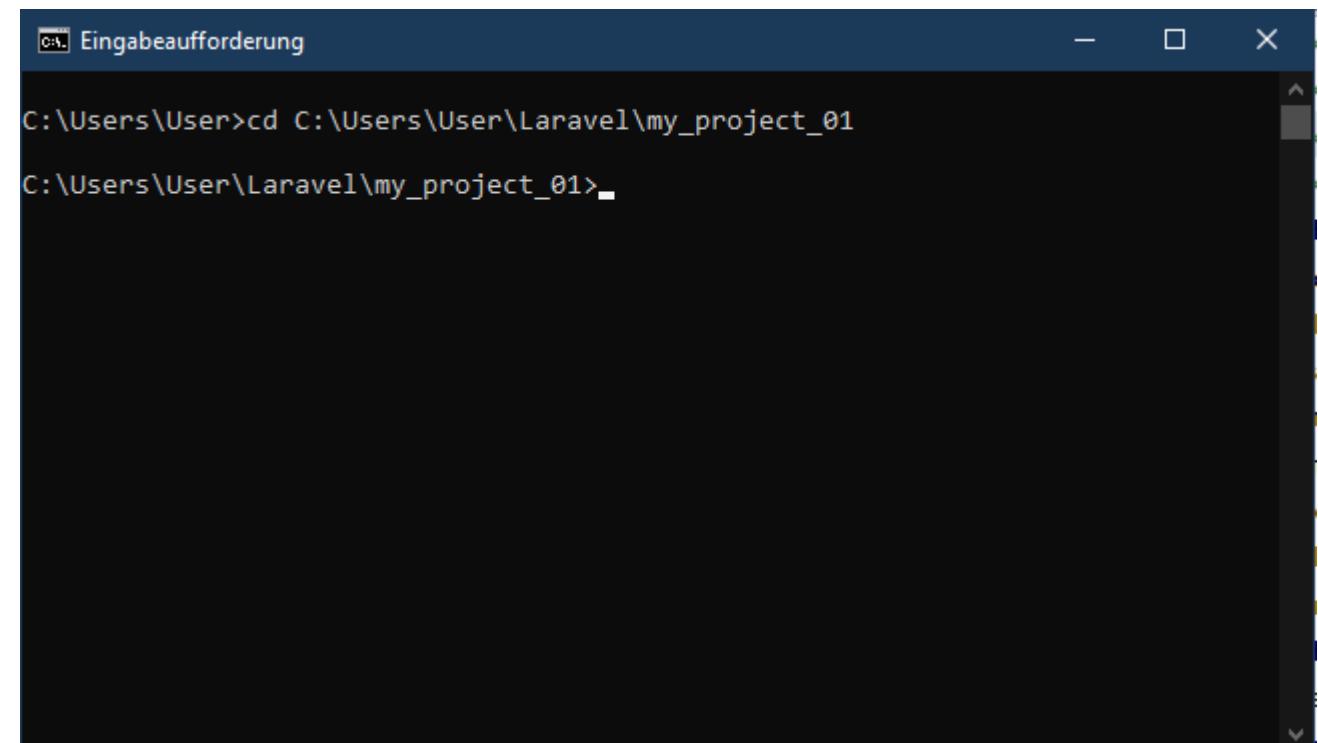
C:\Users\User>
```

cd .. -> einen Ordner nach oben springen

cd C:\Users\User\Laravel\my_project_01

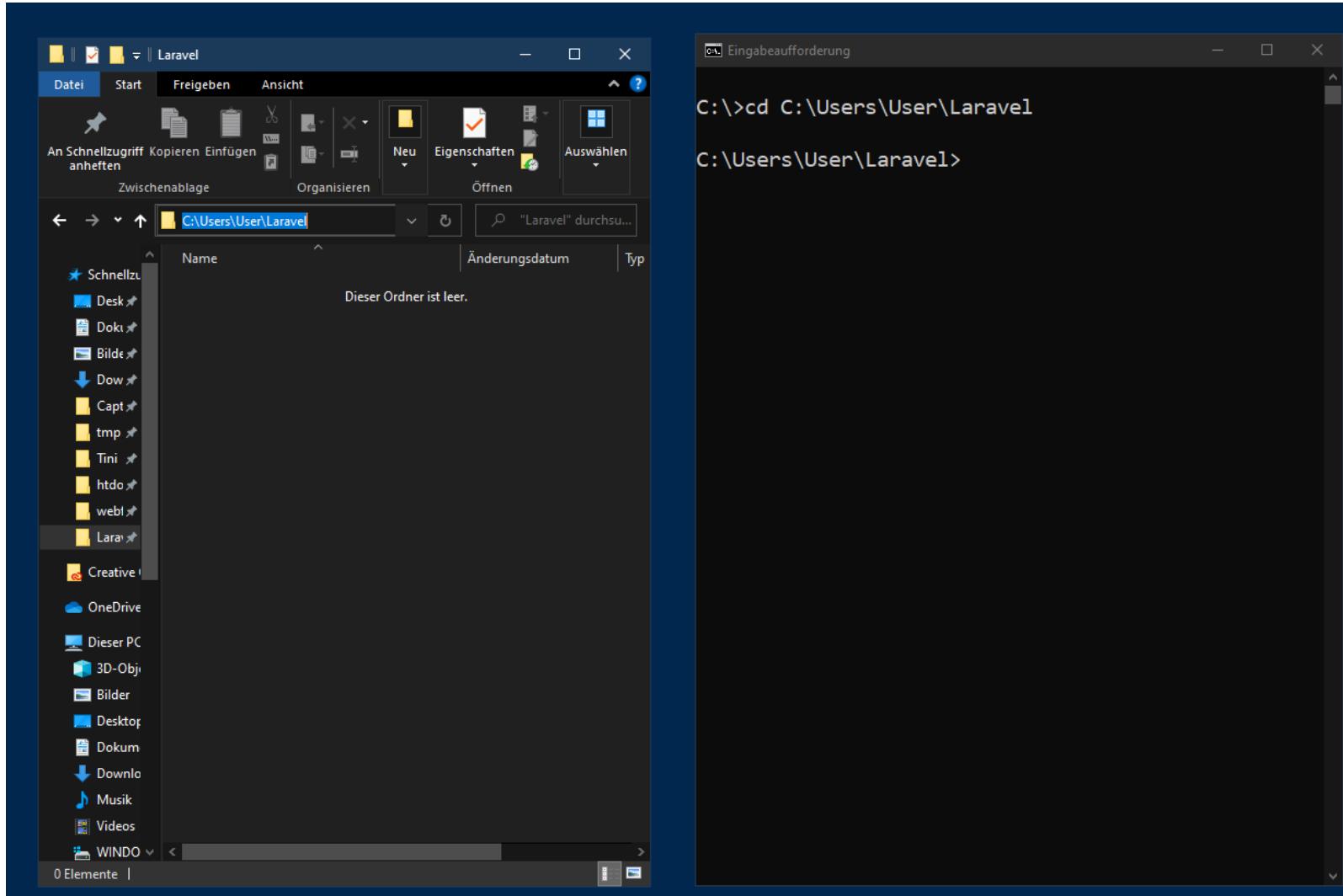


```
Eingabeaufforderung
C:\Users\User>cd ..
C:\Users>cd..
C:\>cd Users
C:\Users>cd user
C:\Users\User>cd \
C:\>
```

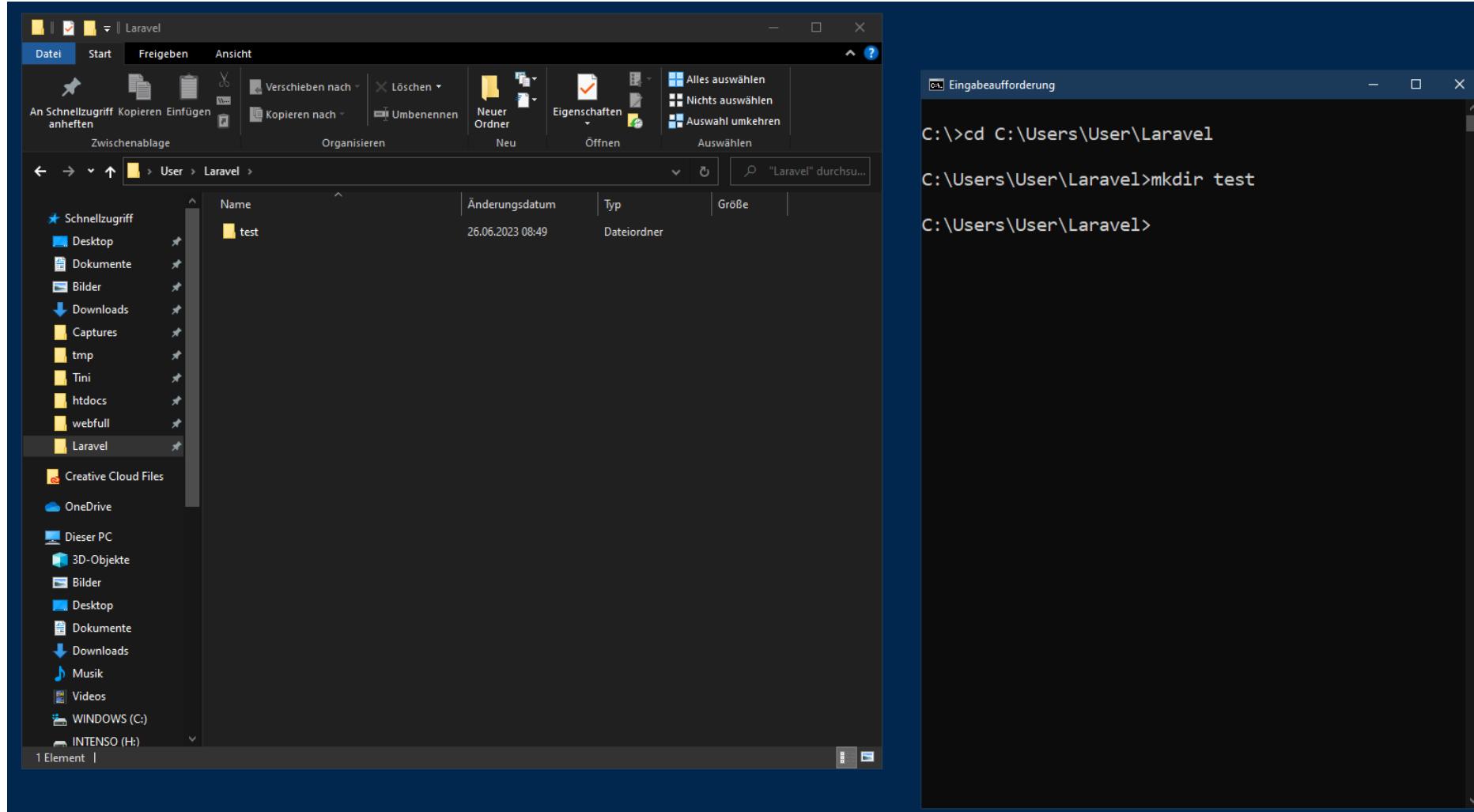


```
Eingabeaufforderung
C:\Users\User>cd C:\Users\User\Laravel\my_project_01
C:\Users\User\Laravel\my_project_01>
```

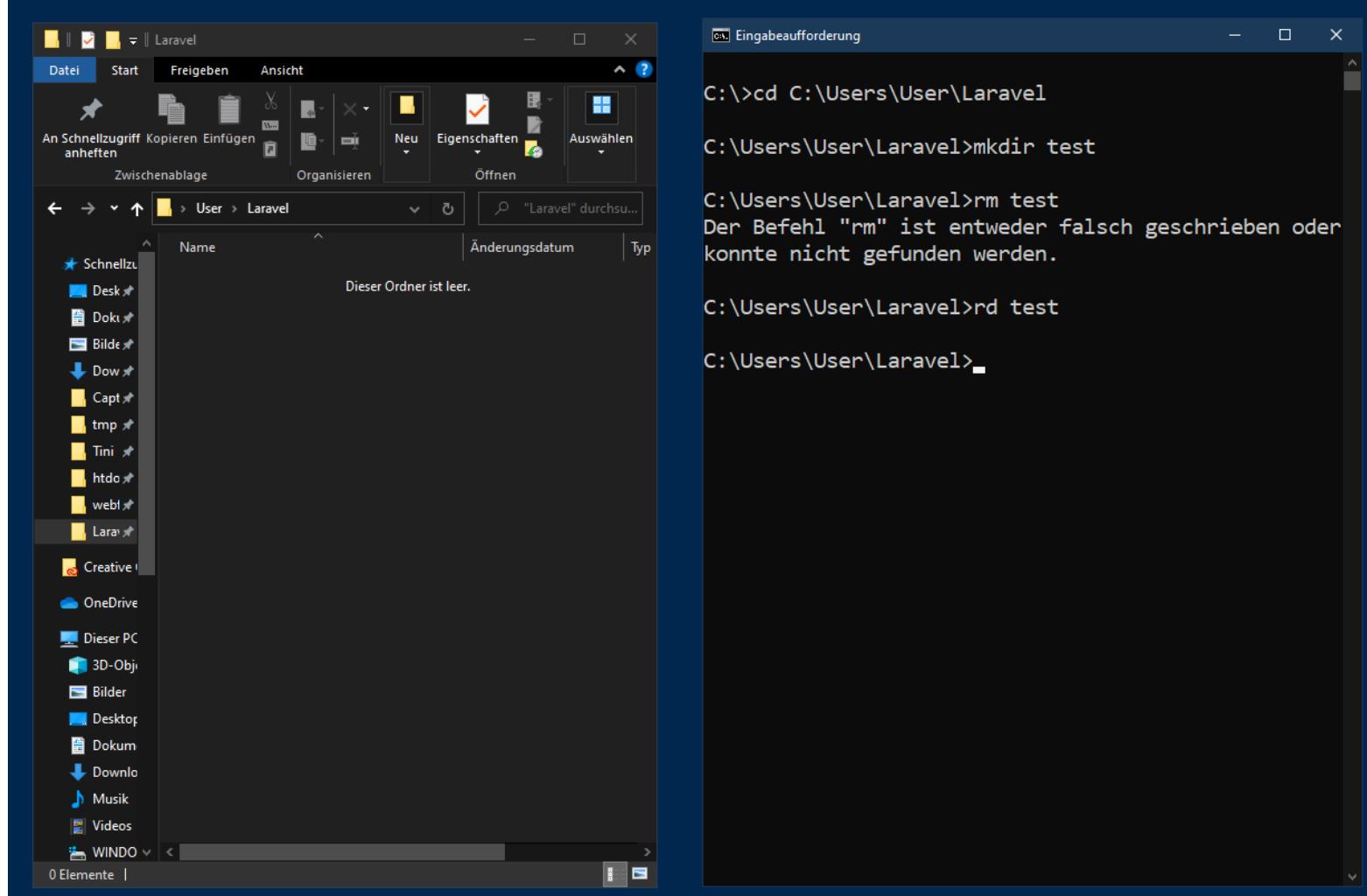
Weg kopieren und einfügen geht auch



Ordner erstellen – mkdir Ordnername



Ordner löschen -> mit rm Ordnername oder rd Ordnername



Ipconfig -> Netzdaten erfragen

```
Eingabeaufforderung

C:\>cd C:\Users\User\Laravel
C:\Users\User\Laravel>ipconfig

Windows-IP-Konfiguration

Ethernet-Adapter Ethernet 3:
  Verbindungsspezifisches DNS-Suffix:
  IPv4-Adresse . . . . . : 192.168.0.170
  Subnetzmaske . . . . . : 255.255.255.0
  Standardgateway . . . . . : 192.168.0.1

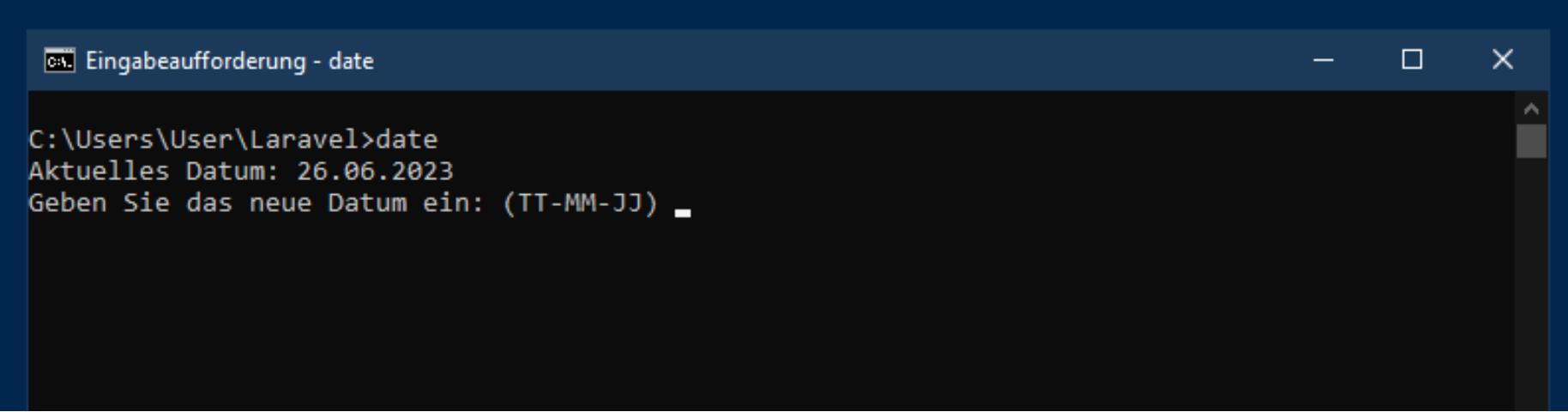
Ethernet-Adapter VirtualBox Host-Only Network:
  Verbindungsspezifisches DNS-Suffix:
  Verbindungslokale IPv6-Adresse . . : fe80::d3f7:a772:fcd:574d%16
  IPv4-Adresse . . . . . : 192.168.56.1
  Subnetzmaske . . . . . : 255.255.255.0
  Standardgateway . . . . . :

Ethernet-Adapter VirtualBox Host-Only Network #2:
  Verbindungsspezifisches DNS-Suffix:
  Verbindungslokale IPv6-Adresse . . : fe80::3f5e:e992:7d33:5998%12
  IPv4-Adresse . . . . . : 192.168.10.1
  Subnetzmaske . . . . . : 255.255.255.0
  Standardgateway . . . . . :

Ethernet-Adapter Ethernet 4:
  Medienstatus. . . . . . . : Medium getrennt
  Verbindungsspezifisches DNS-Suffix:

C:\Users\User\Laravel>
```

date -> Datum erfragen



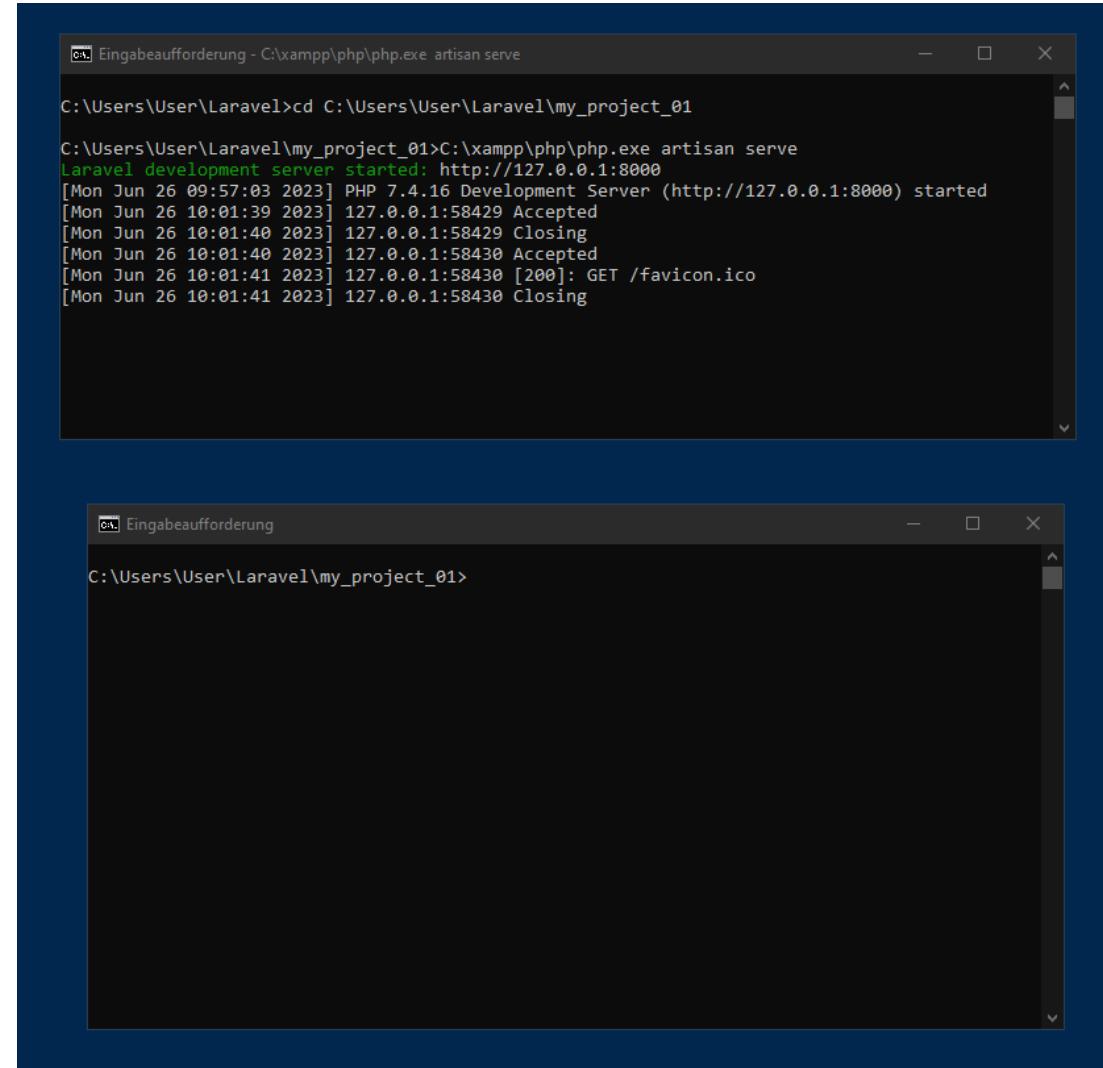
```
Eingabeaufforderung - date
C:\Users\User\Laravel>date
Aktuelles Datum: 26.06.2023
Geben Sie das neue Datum ein: (TT-MM-JJ) -
```

Erste Schritte um Laravel zu starten

Man braucht 2 Fenster!

-Oben: startet Projekt und muss offen bleiben

-Unten: Programmiere ich dann



The image displays two separate command-line windows, likely from a Windows operating system, illustrating a workflow for developing a Laravel application.

The top window, titled "Eingabeaufforderung - C:\xampp\php\php.exe artisan serve", shows the output of running the Artisan serve command. It outputs the following text:

```
C:\Users\User\Laravel>cd C:\Users\User\Laravel\my_project_01
C:\Users\User\my_project_01>C:\xampp\php\php.exe artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Jun 26 09:57:03 2023] PHP 7.4.16 Development Server (http://127.0.0.1:8000) started
[Mon Jun 26 10:01:39 2023] 127.0.0.1:58429 Accepted
[Mon Jun 26 10:01:40 2023] 127.0.0.1:58429 Closing
[Mon Jun 26 10:01:40 2023] 127.0.0.1:58430 Accepted
[Mon Jun 26 10:01:41 2023] 127.0.0.1:58430 [200]: GET /favicon.ico
[Mon Jun 26 10:01:41 2023] 127.0.0.1:58430 Closing
```

The bottom window, also titled "Eingabeaufforderung", shows the prompt "C:\Users\User\Laravel\my_project_01>" indicating it is ready for further input.

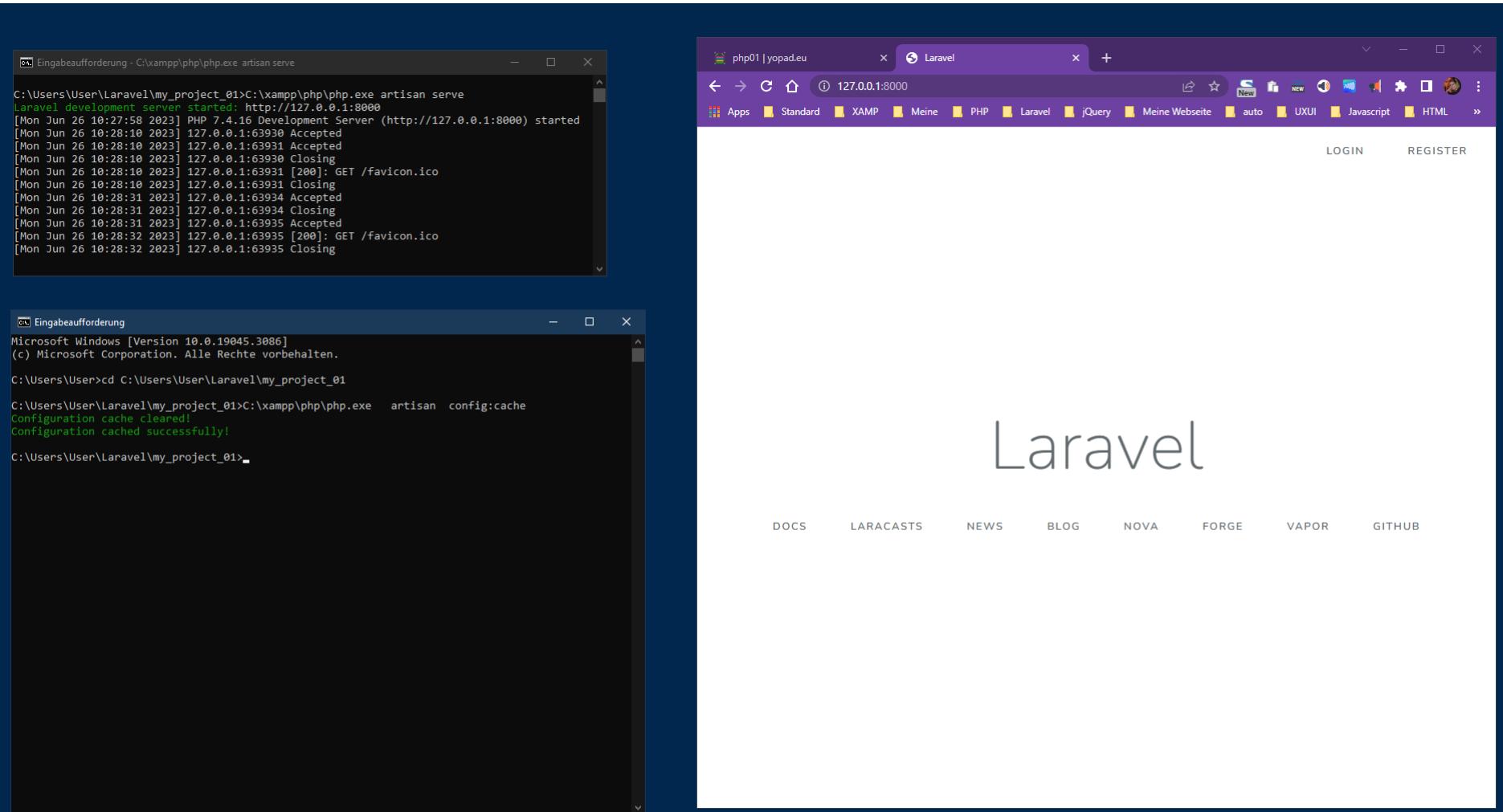
Erstes Fenster / oben: Startet Laravel

- Bei anderen Rechnern geht: php artisan serve
- Bei uns hier: C:\xampp\php\php.exe artisan serve

```
Eingabeaufforderung - C:\xampp\php\php.exe artisan serve
C:\Users\User\Laravel>cd C:\Users\User\Laravel\my_project_01
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Jun 26 09:57:03 2023] PHP 7.4.16 Development Server (http://127.0.0.1:8000) started
```

Zweites Fenster - Löscht Speicher

- Bei anderen Rechnern : php artisan config:cache
- Bei uns: C:\xampp\php\php.exe artisan config:cache



The image displays three windows. The top-left window is a command-line interface showing the output of running `artisan serve`, which starts a Laravel development server at `http://127.0.0.1:8000`. The top-right window is a web browser showing the Laravel homepage at `127.0.0.1:8000`. The bottom-left window is another command-line interface showing the output of running `artisan config:cache`, which clears and caches the configuration.

```
Eingabeaufforderung - C:\xampp\php\php.exe artisan serve
C:\Users\User\laravel\my_project_01>C:\xampp\php\php.exe artisan serve
Laravel development server started: http://127.0.0.1:8000
[Mon Jun 26 10:27:58 2023] PHP 7.4.16 Development Server (http://127.0.0.1:8000) started
[Mon Jun 26 10:28:10 2023] 127.0.0.1:63930 Accepted
[Mon Jun 26 10:28:10 2023] 127.0.0.1:63931 Accepted
[Mon Jun 26 10:28:10 2023] 127.0.0.1:63930 Closing
[Mon Jun 26 10:28:10 2023] 127.0.0.1:63931 [200]: GET /favicon.ico
[Mon Jun 26 10:28:10 2023] 127.0.0.1:63931 Closing
[Mon Jun 26 10:28:31 2023] 127.0.0.1:63934 Accepted
[Mon Jun 26 10:28:31 2023] 127.0.0.1:63934 Closing
[Mon Jun 26 10:28:31 2023] 127.0.0.1:63935 Accepted
[Mon Jun 26 10:28:32 2023] 127.0.0.1:63935 [200]: GET /favicon.ico
[Mon Jun 26 10:28:32 2023] 127.0.0.1:63935 Closing

Eingabeaufforderung
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\User>cd C:\Users\User\laravel\my_project_01

C:\Users\User\laravel\my_project_01>C:\xampp\php\php.exe artisan config:cache
Configuration cache cleared!
Configuration cached successfully!

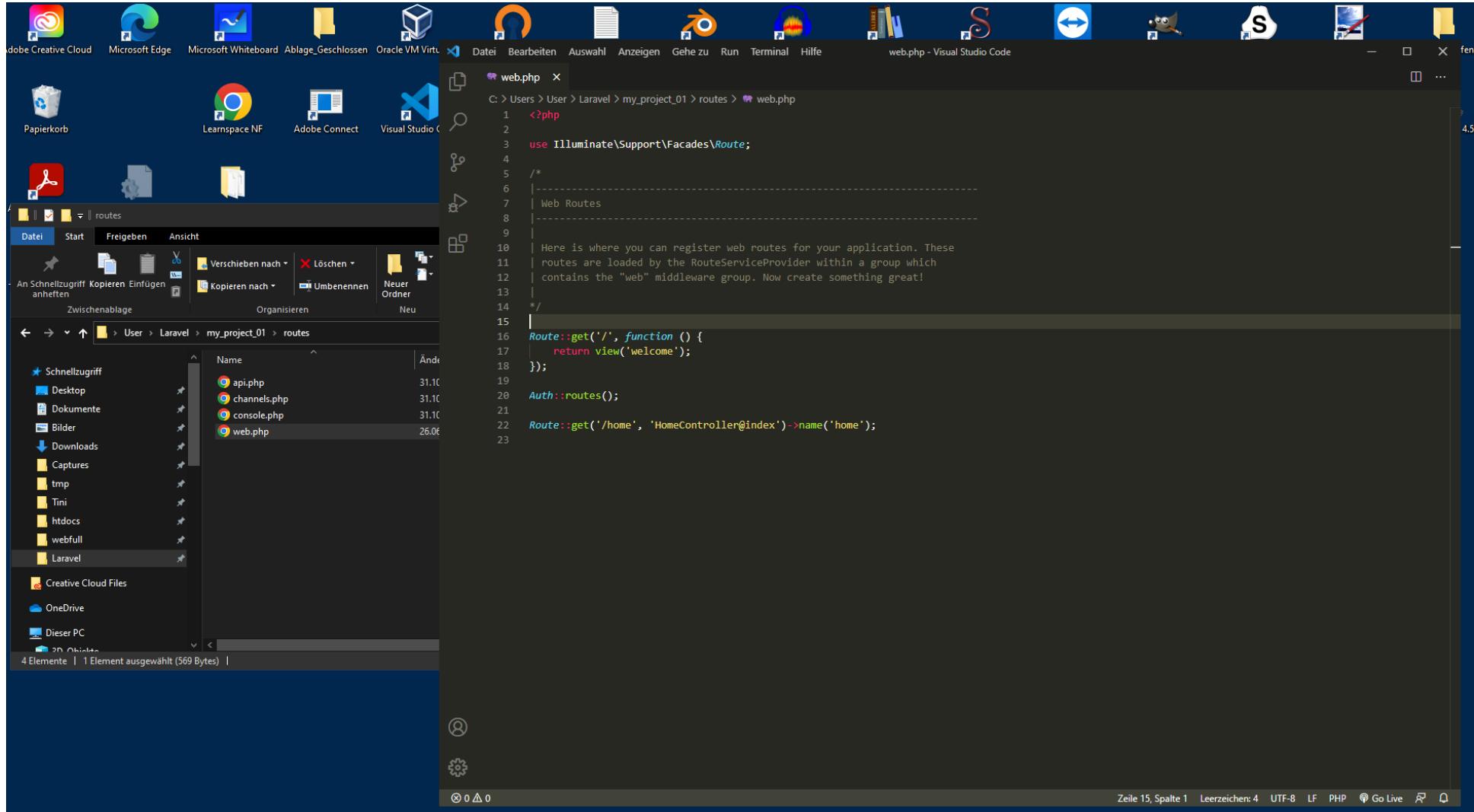
C:\Users\User\laravel\my_project_01>
```

Laravel

DOCS LARACASTS NEWS BLOG NOVA FORGE VAPOR GITHUB

Einige Seiten kurz
erklärt

web.php - wichtige Datei - steuert Navi oben und Bindet Seiten ein



The screenshot shows a Windows desktop environment with a dark theme. In the background, the Start menu is open, displaying various pinned icons like Adobe Creative Cloud, Microsoft Edge, Microsoft Whiteboard, and Oracle VM VirtualBox. In the foreground, Visual Studio Code is running, showing the file `routes/web.php`. The code defines web routes for a Laravel application:

```
<?php  
use Illuminate\Support\Facades\Route;  
  
/*  
| Web Routes  
|  
| Here is where you can register web routes for your application. These  
| routes are loaded by the RouteServiceProvider within a group which  
| contains the "web" middleware group. Now create something great!  
*/  
  
Route::get('/', function () {  
    return view('welcome');  
});  
  
Auth::routes();  
  
Route::get('/home', 'HomeController@index')->name('home');
```

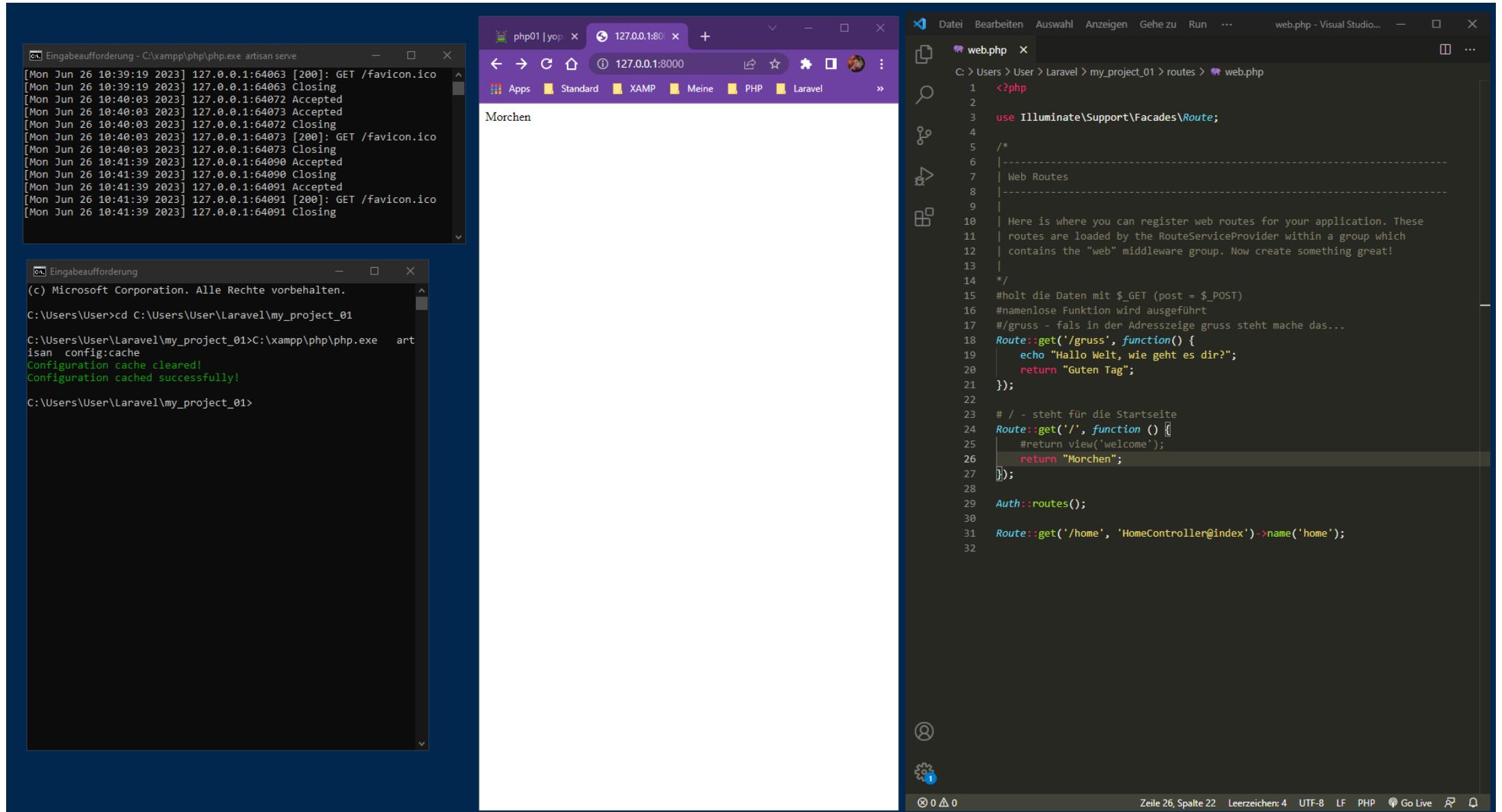
The left sidebar of VS Code shows the file structure of the Laravel project, with files like `api.php`, `channels.php`, `console.php`, and `web.php` listed under the `routes` directory.

web.php bearbeiten

The screenshot shows a Windows desktop environment with three windows open:

- Terminal Window (Bottom Left):** Shows command-line output from running `artisan serve`. It lists multiple log entries from June 26, 2023, at 10:40:03, indicating requests for favicon.ico and the root route, along with responses like "Accepted" and "Closing".
- Browser Window (Top Middle):** Displays a simple web page with the text "Hallo Welt, wie geht es dir? Guten Tag". The URL in the address bar is `127.0.0.1:8000/gruss`.
- Code Editor Window (Right):** Shows the contents of the `web.php` file in a Visual Studio Code editor. The code defines routes for the "gruss" and root paths, and includes a call to `Auth::routes()`. A specific line of code is highlighted: `Route::get('/', function () { echo "Hallo Welt, wie geht es dir?"; return "Guten Tag";});`.

web.php bearbeiten



The screenshot shows a Windows desktop environment with three main windows open:

- Terminal Window (Left):** Displays command-line output from XAMPP's php.exe artisan serve. It shows log entries for various requests, including several favicon.ico requests.
- Browser Window (Middle):** Shows a simple web page with the text "Morchen". The browser address bar indicates the URL is 127.0.0.1:8000.
- Visual Studio Code Window (Right):** Displays the contents of the 'web.php' file. The code defines routes for 'gruss' and 'home'.

```
<?php
use Illuminate\Support\Facades\Route;
/*
|--------------------------------------------------------------------------
| Web Routes
|--------------------------------------------------------------------------
|
| Here is where you can register web routes for your application. These
| routes are loaded by the RouteServiceProvider within a group which
| contains the "web" middleware group. Now create something great!
|
*/
# holte die Daten mit $_GET (post = $_POST)
# namenlose Funktion wird ausgeführt
# gruss - fals in der Adresszeile gruss steht macht das...
Route::get('/gruss', function () {
    echo "Hallo Welt, wie geht es dir?";
    return "Guten Tag";
});

# / - steht für die Startseite
Route::get('/', function () {
    #return view('welcome');
    return "Morchen";
});

Auth::routes();

Route::get('/home', 'HomeController@index')->name('home');
```

Visual Studio Code status bar at the bottom: Zeile 26, Spalte 22 Leerzeichen: 4 UTF-8 LF PHP Go Live

Seiten mit xxx.blade.php bestimmen den Inhalt
der Seiten

```
Eingabeaufforderung - C:\xampp\php\php.exe artisan serve
[Mon Jun 26 10:46:14 2023] 127.0.0.1:64129 Accepted
[Mon Jun 26 10:46:14 2023] 127.0.0.1:64129 [200]: GET /favicon.ico
[Mon Jun 26 10:46:14 2023] 127.0.0.1:64129 Closing
[Mon Jun 26 10:47:17 2023] 127.0.0.1:64139 Accepted
[Mon Jun 26 10:47:17 2023] 127.0.0.1:64139 Closing
[Mon Jun 26 10:47:17 2023] 127.0.0.1:64140 Accepted
[Mon Jun 26 10:47:17 2023] 127.0.0.1:64140 [200]: GET /favicon.ico
[Mon Jun 26 10:47:17 2023] 127.0.0.1:64140 Closing
[Mon Jun 26 10:47:37 2023] 127.0.0.1:64145 Accepted
[Mon Jun 26 10:47:37 2023] 127.0.0.1:64146 Accepted
[Mon Jun 26 10:47:37 2023] 127.0.0.1:64145 Closing
[Mon Jun 26 10:47:37 2023] 127.0.0.1:64146 [200]: GET /favicon.ico
[Mon Jun 26 10:47:37 2023] 127.0.0.1:64146 Closing

Eingabeaufforderung
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\User>cd C:\Users\User\Laravel\my_project_01

C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan config:cache
Configuration cache cleared!
Configuration cached successfully!

C:\Users\User\Laravel\my_project_01>
```



Laravel

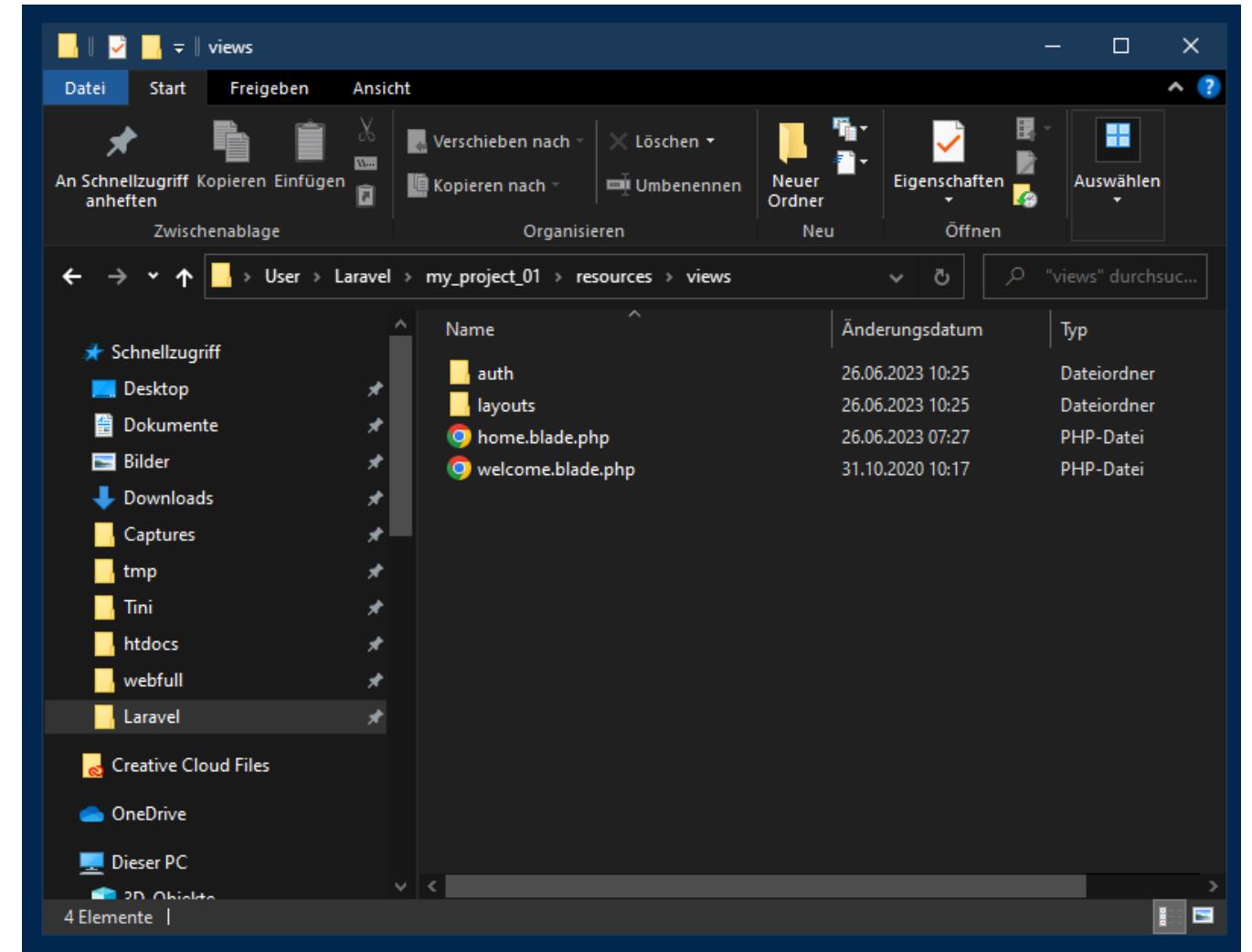
Laravel ist einfach

```
麒 Datei Bearbeiten Auswahl Anzeigen Gehe zu Run ... welcome.blade.php - Visual St... — □ ...  
麒 web.php麒 welcome.blade.php x  
C:\Users\User>Laravel>my_project_01>resources>views>welcome.blade.php  
麒 62                          margin-bottom: 30px;  
麒 63        }  
麒 64    </style>  
麒 65 </head>  
麒 66 <body>  
麒 67        <div class="flex-center position-ref full-height">  
麒 68            @if (Route::has('login'))  
麒 69                <div class="top-right links">  
麒 70                    @auth  
麒 71                        <a href="{{ url('/home') }}>Home</a>  
麒 72                    @else  
麒 73                        <a href="{{ route('login') }}>Login</a>  
麒 74  
麒 75                    @endif  
麒 76                    @endauth  
麒 77                </div>  
麒 78        @endif  
麒 79        <div class="content">  
麒 80            <div class="title m-b-md">  
麒 81                Laravel  
麒 82                <p>Laravel ist einfach!</p>  
麒 83            </div>  
麒 84  
麒 85        <div class="links">  
麒 86            <a href="https://laravel.com/docs">Docs</a>  
麒 87            <a href="https://laracasts.com">Laracasts</a>  
麒 88            <a href="https://laravel-news.com">News</a>  
麒 89            <a href="https://blog.laravel.com">Blog</a>  
麒 90            <a href="https://nova.laravel.com">Nova</a>  
麒 91            <a href="https://forge.laravel.com">Forge</a>  
麒 92            <a href="https://vapor.laravel.com">Vapor</a>  
麒 93            <a href="https://github.com/laravel/laravel">GitHub</a>  
麒 94  
麒 95        </div>  
麒 96        </div>  
麒 97    </body>  
麒 98 </html>  
麒 99  
麒 100  
麒 101  
麒 102
```

Hier werden die Seiten abgelegt

xxx.blade.php

– Seiten müssen so geschrieben werden, damit Laravel das Erkennt



Befehle - Seiteninhalt

Laravel hat ein @davor

(Direktive von Laravel)

HTML wie gewohnt

php wie gewohnt



```
startseite.blade.php ×
my_project_01 > resources > views > startseite.blade.php
1  @extends('layouts.app')
2
3  <!-- section beginnt hier -->
4  @section('title', 'Startseite')
5
6
7
8
9  <!-- section beginnt hier -->
10 @section('content')
11 <div class="container">
12   <div class="row">
13     <div class="col-md-12">
14       <div class="card">
15         <div class="card-header">
16           Startseite
17         </div>
18         <div class="card-body">
19           <p>
20             Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.
21           </p>
22         </div>
23       </div>
24     </div>
25   </div>
26 </div>
27 <!-- section endet hier -->
28 @endsection
29
```

Einbinden durch Laravelbefehle

The screenshot shows two open files in Visual Studio Code:

- startseite.blade.php** (Left Tab):

```
1 @extends('layouts.app')
2
3 <!-- section beginnt hier -->
4 @section('content')
5 <div class="container">
6   <div class="row">
7     <div class="col-md-12">
8       <div class="card">
9         <div class="card-header">
10          Startseite
11        </div>
12        <div class="card-body">
13          <p>
14            Lorem ipsum dolor sit amet, consetetur
15            sadipscing elitr, sed diam nonumy eirmod tempor
16            invidunt ut labore et dolore magna aliquyam erat,
17            sed diam voluptua. At vero eos et accusam etiam
18            ipsum圣人所欲也，非有能也，唯无能者能之。
19          </p>
20        </div>
21      </div>
22    </div>
23  </div>
24 </div>
25 <!-- section endet hier -->
26 @endsection
```
- app.blade.php** (Right Tab):

```
72           </li>
73           @endguest
74         </ul>
75       </div>
76     </div>
77   </nav>
78
79   <main class="py-4">
80     @yield('content')
81   </main>
82 </div>
83 </body>
84 </html>
85
```

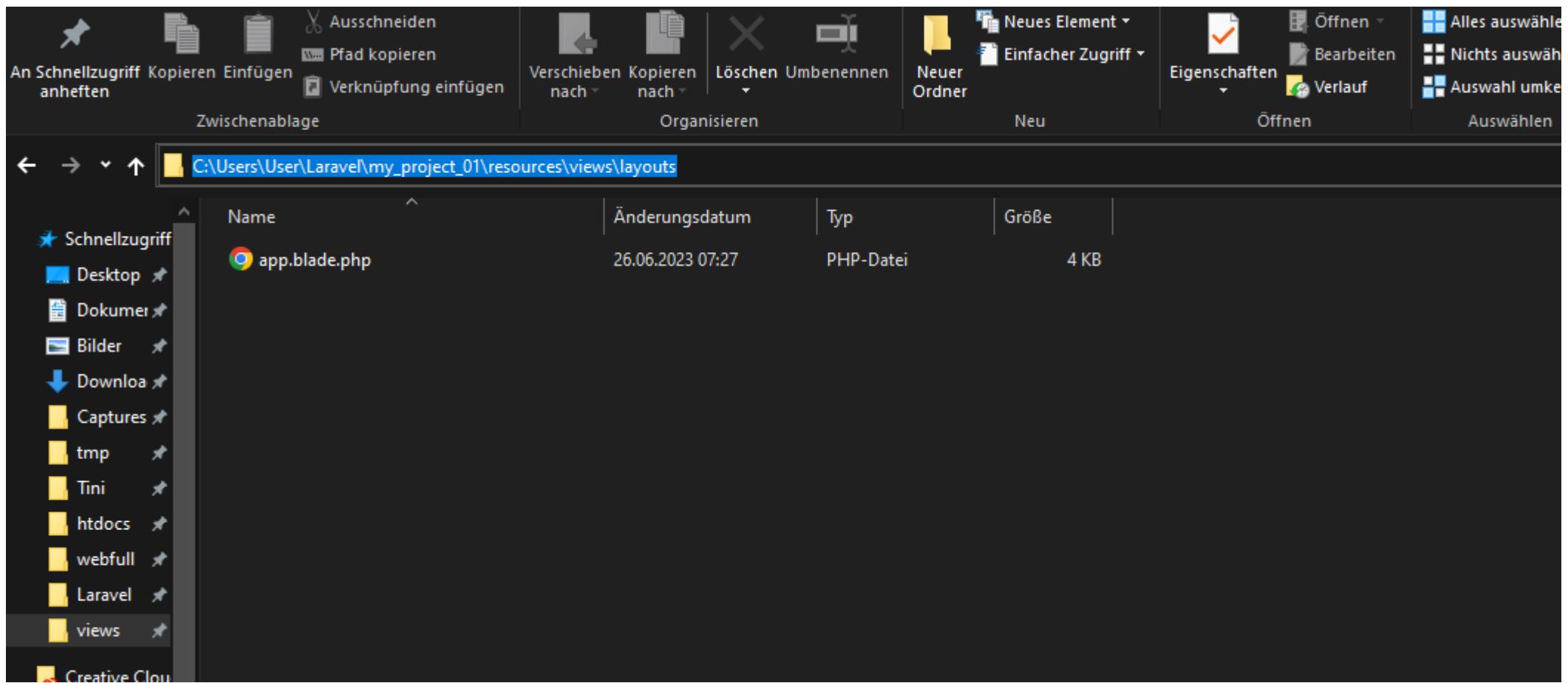
@yield('xxxxxx') ist container und @section ('xxxxxx') zeigt was in @yield ('xxxxxx') angezeigt werden soll

@extends('layouts.app') – sagt das @section dort eingefügt werden soll

layout.app – ist Datei

layout.app. – ist Ordner

Wichtige Datei – steuert das Einbinden der Seiten



Navigation
ändern/erweitern

Navigation für meine Seiten einbinden

The screenshot shows a Windows desktop environment with several open windows:

- Terminal Window:** Shows command-line output for Laravel configuration caching, indicating "Configuration cached successfully!"
- Browser Window:** Displays a Laravel login page with fields for "E-Mail Address" and "Password", and links for "Remember Me", "Login", and "Forgot Your Password?".
- Code Editor:** Visual Studio Code is open, displaying the file `app.blade.php`. The code is a Blade template for a navigation bar. A specific section of the code is highlighted in gray, corresponding to the menu items in the browser's navbar.
- File Explorer:** Shows the project structure with folders like "layouts", "views", and files such as "app.blade.php".

```
17 f="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
18
19 @vite('resources/css/app.css') rel="stylesheet"
20
21
22 <!-- Left Side Of Navbar -->
23 <ul class="navbar-nav mr-auto">
24   <li><a href="/">Startseite</a></li>
25   <li><a href="/Über uns">Über uns</a></li>
26   <li><a href="/Meine Referenzen">Meine Referenzen</a></li>
27   <li><a href="/Kontakt">Kontakt</a></li>
28 </ul>
29
30 <!-- Right Side Of Navbar -->
31 <ul class="navbar-nav ml-auto">
32   <li><a href="{{ route('login') }}>{{ __('Login') }}</a></li>
33   @if(Route::has('register'))
34     <li><a href="{{ route('register') }}>{{ __('Register') }}</a></li>
35   @endif
36   @else
37     <li><a href="#" id="navbarDropdown" class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">{{ Auth::user()->name }}</a></li>
38   <div class="dropdown-menu dropdown-menu-right" aria-labelledby="navbarDropdown">
39     <a href="{{ route('logout') }}" class="dropdown-item" href="#" onclick="event.preventDefault();
40       document.getElementById('logout-form').submit();">{{ __('Logout') }}</a>
41   </div>
42 </ul>
```

Bestimmen wenn bestimmter Begriff in der Adresszeile - dann folgende Datei anzeigen

The screenshot illustrates the configuration of a Laravel application to serve specific content based on the URL path.

Browser View: The browser shows a Laravel application at `127.0.0.1:8000/adresse`. The page displays a header with "Laravel" and navigation links: Startseite, Über uns, Meine Referenzen, Kontakt, Login, Register. Below the header is a main content area titled "Kontakt - So können Sie uns erreichen". The content area contains placeholder text (Lorem ipsum) and a red arrow points from the "Kontakt" link in the browser's address bar to this content area.

VS Code Editor: The editor displays two files:

- `routes/web.php`: Contains the following code:

```
use Illuminate\Support\Facades\Route;

Route::get('/gruss', function () {
    echo "Hallo Welt, wie geht es dir?";
    return "Guten Tag";
});

Route::get('/', function () {
    return view('startseite');
});

Route::get('/about', function () {
    return view('about');
});

Route::get('/referenzen', function () {
    return view('referenzen');
});

Route::get('/adresse', function () {
    return view('kontakt');
});

Auth::routes();
```
- `resources/views/app.blade.php`: Contains the following code:

```
<?php

$container = new \Illuminate\Support\Arr\Arr();

$container['app.name'] = 'Laravel';

$container->bind('app.name', function () {
    return $app->environment();
});
```

A red arrow points from the "adresse" link in the browser's address bar to the corresponding route definition in `routes/web.php`.

Bottom Status Bar: The status bar at the bottom of the VS Code window shows: Zeile 42, Spalte 1, Leerzeichen: 4, UTF-8, LF, PHP, Go Live.

Tabnamen (Titel) ändern

The screenshot illustrates how to change the tab title in a browser by modifying the page's title in the blade template.

Browser Tab: The browser tab is titled "myProject". A red arrow points from this tab to the code editor.

Code Editor: The code editor shows two files: "web.php" and "app.blade.php".

- web.php:** This file contains the following code:

```
C: > Users > User > Laravel > my_project_01 > resources > views > web.php
1 @extends('layouts.app')
2
3 <!-- section beginnt hier -->
4 @section('title','Startseite')
5
6
7
8
9 <!-- section beginnt hier -->
10 @section('content')
11 <div class="container">
12   <div class="row">
13     <div class="col-md-12">
14       <div class="card">
15         <div class="card-header">
16           Startseite
17         </div>
18         <div class="card-body">
19           <p>Lorem ipsum dolor sit amet</p>
20         </div>
21       </div>
22     </div>
23   </div>
24 </div>
25 </div>
26 <!-- section endet hier -->
27 @endsection
28
```
- app.blade.php:** This file contains the following code:

```
C: > Users > User > Laravel > my_project_01 > resources > views > layouts > app.blade.php
1 <!doctype html>
2 <html lang="{{ str_replace('_', '-', app()->getLocale()) }}">
3   <head>
4     <meta charset="utf-8">
5     <meta name="viewport" content="width=device-width, initial-scale=1">
6
7     <!-- CSRF Token -->
8     <meta name="csrf-token" content="{{ csrf_token() }}>
9
10    <title> @yield['title','myProject']</title>
11
12    <!-- Scripts -->
13    <script src="{{ asset('js/app.js') }}" defer></script>
14
15    <!-- Fonts -->
16    <link rel="dns-prefetch" href="https://fonts.gstatic.com">
17    <link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">
18
19    <!-- Styles -->
20    <link href="{{ asset('css/app.css') }}" rel="stylesheet">
21
22  </head>
23  <body>
24    <div id="app">
25      <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
26        <div class="container">
27          <a class="navbar-brand" href="{{ url('/') }}>
28            {{ config('app.name', 'Laravel') }}</a>
29          </a>
30          <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" data-aos="fade-right">
31            <span class="navbar-toggler-icon"></span>
32          </button>
33
34          <div class="collapse navbar-collapse" id="navbarSupportedContent">
35            <!-- Left Side Of Navbar -->
36            <ul class="navbar-nav mr-auto">
37              <!-- / steht für Startseite -->
38              <li> <a class="nav-link" href="/">Startseite</a></li>
39              <li> <a class="nav-link" href="/about">Über uns</a></li>
40              <li> <a class="nav-link" href="/referenzen">Meine Referenzen</a></li>
41              <li> <a class="nav-link" href="/adresse">Kontakt</a></li>
42            </ul>
43          </div>
44        </div>
45      </nav>
46      <div class="container">
47        <div class="row">
48          <div class="col-md-12">
49            <div class="card">
50              <div class="card-header">
51                Startseite
52              </div>
53              <div class="card-body">
54                <p>Lorem ipsum dolor sit amet</p>
55              </div>
56            </div>
57          </div>
58        </div>
59      </div>
60    </div>
61  </body>
62</html>
```

Projektnamenanzeige ändern

- geht mit Logo und Text

The screenshot illustrates the process of changing the project name from 'myProject' to 'MyProject'. On the left, a browser window shows a website with the title 'MyProject' in the header. A red arrow points to the word 'MyProject' in the header. On the right, a code editor displays the file 'app.blade.php'. A second red arrow points to the word 'MyProject' in the code, specifically within the ` MyProject ` line.

```
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- CSRF Token -->
<meta name="csrf-token" content="{{ csrf_token() }}>

<title> @yield('title','myProject')</title>

<!-- Scripts -->
<script src="{{ asset('js/app.js') }}" defer></script>

<!-- Fonts -->
<link rel="dns-prefetch" href="//fonts.gstatic.com">
<link href="https://fonts.googleapis.com/css?family=Nunito" rel="stylesheet">

<!-- Styles -->
<link href="{{ asset('css/app.css') }}" rel="stylesheet">

</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}> MyProject </a>
                <!-- {{ config('app.name', 'Laravel') }} -->
            </a>
            <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" data-bs-toggle="button" data-bs-target="#navbarSupportedContent">
                <span class="navbar-toggler-icon"></span>
            </button>
            <div class="collapse navbar-collapse" id="navbarSupportedContent">
                <ul class="navbar-nav me-auto mb-2 mb-lg-0">
                    <li class="nav-item">
                        <a class="nav-link" href="#">Startseite
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Über uns
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Meine Referenzen
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Kontakt
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Login
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Register
                    </li>
                </ul>
            </div>
        </nav>
    </div>
</body>

```

Navi hervorheben wo ich bin

The screenshot illustrates how to highlight the current page in a navigation bar using Laravel's Blade templating engine.

Code Editor View: Shows the `app.blade.php` template with the following navigation bar code:

```
<!-- Left Side Of Navbar -->
<ul class="navbar-nav mr-auto">
    <!-- / steht für Startseite -->
    <!--request spricht inhalt an-->
    <li> <a class="nav-link {{ Request::is('/') ? 'btn bg-success' : ''}}" href="/">Startseite</a></li>
    <li> <a class="nav-link {{ Request::is('about') ? 'btn bg-success' : ''}}" href="/about">Über uns</a></li>
    <li> <a class="nav-link {{ Request::is('referenzen') ? 'btn bg-success' : ''}}" href="/referenzen">Meine Referenzen</a></li>
    <li> <a class="nav-link {{ Request::is('post*') ? 'btn bg-success' : ''}}" href="/post">Beiträge</a></li>
    <li> <a class="nav-link {{ Request::is('adresse') ? 'btn bg-success' : ''}}" href="/adresse">Kontakt</a></li>
</ul>
```

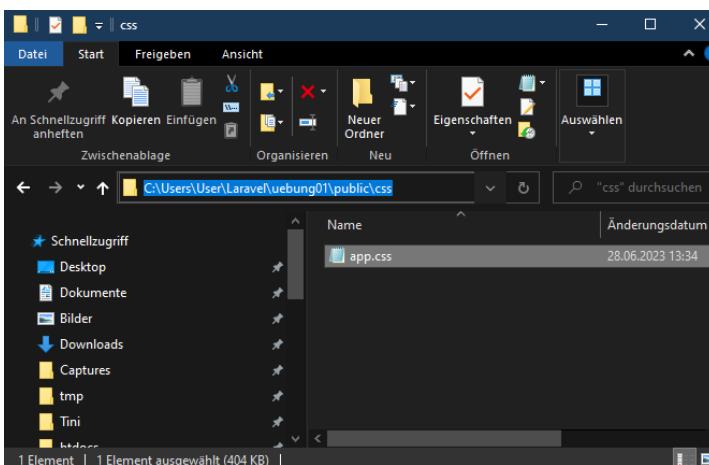
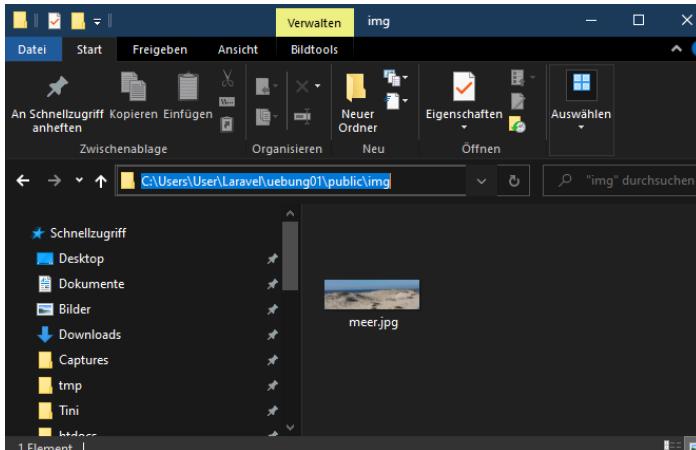
Browser 1 (Left): Shows the homepage (`127.0.0.1:8000`) with the navigation bar. The "Startseite" link is highlighted in green, indicating it is the active page.

Browser 2 (Right): Shows a page titled "Meine Referenzen" (`127.0.0.1:8000/referenzen`). The "Meine Referenzen" link in the navigation bar is highlighted in green, indicating it is the active page.

```
<li> <a class="nav-link {{ Request::is('/') ? 'btn bg-success' : ''}}" href="/">Startseite</a></li>
<li> <a class="nav-link {{ Request::is('adresse') ? 'btn bg-success' : ''}}" href="/adresse">Kontakt</a></li>
```

Hintergrundbild
einbinden

Hintergrundbild einbinden



The image shows a web browser displaying a website with a dark blue header and a background image of a sandy beach with low-lying vegetation. The header includes a navigation bar with 'myProject' and other links like 'Home', 'About', 'Portfolio', 'Kontakt', 'Wert', 'Zahl', and 'User'. Below the header, there's a large yellow text 'webdesign.' followed by 'creativ designer'. A section titled 'design follows function' contains placeholder text. Another section titled 'Kurze Info über mich...' also contains placeholder text. Buttons for 'Mein Portfolio', 'About me', and 'Kontakt' are visible. The overall aesthetic is modern and minimalist.

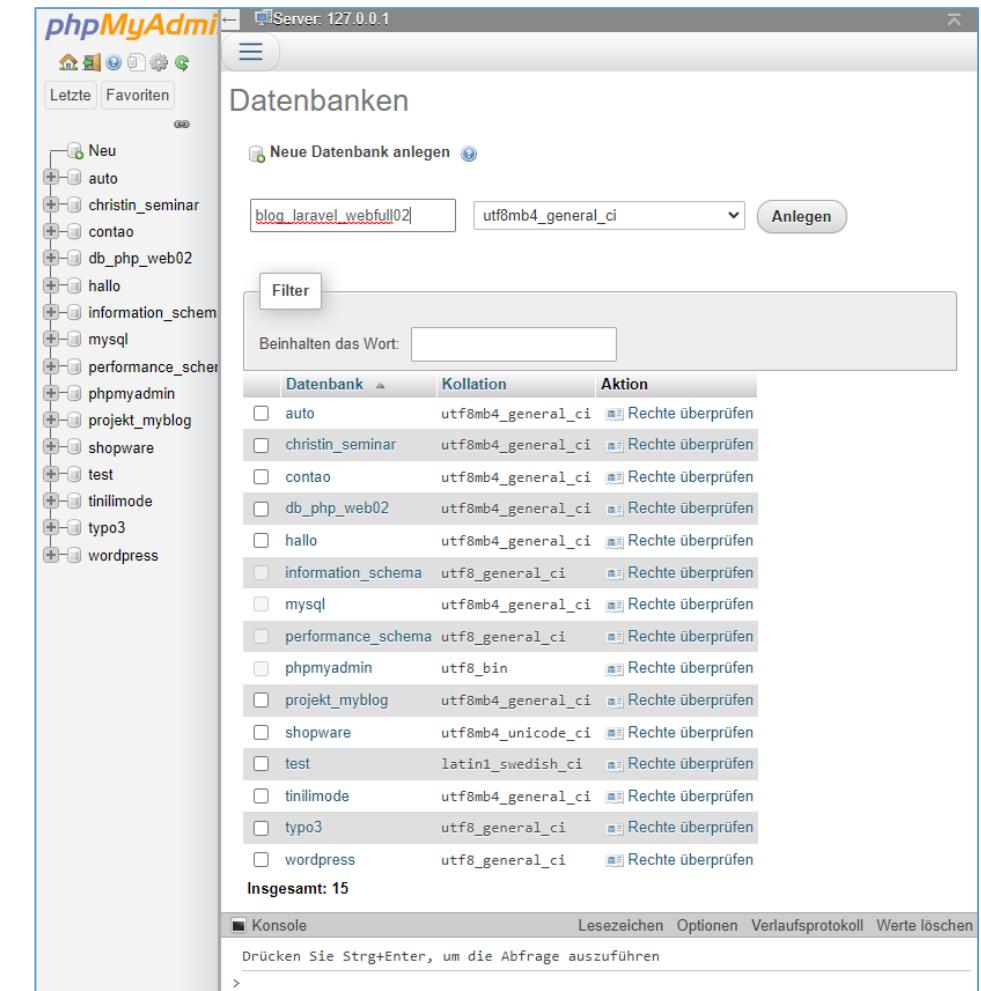
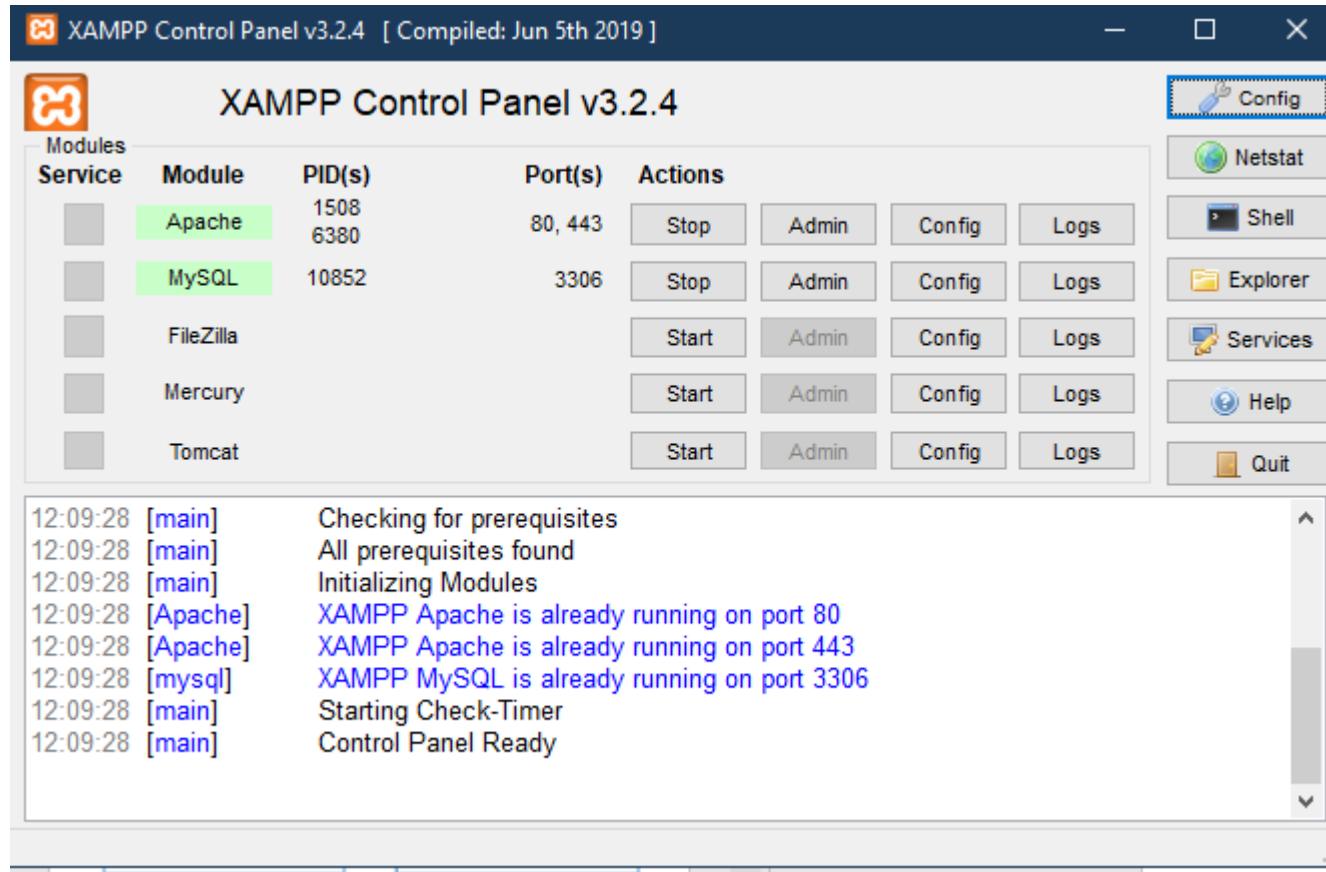
```
*app.css - Editor
Datei Bearbeiten Format Ansicht Hilfe
body {
    background: var(--bs-info) url('/img/meer.jpg') no-repeat center fixed;
    background-size: cover;
}

@import url(https://fonts.googleapis.com/css?family=Nunito);/*
 * Font Awesome Free 6.4.0 by @fontawesome - https://fontawesome.com
 * License - https://fontawesome.com/license/free (Icons: CC BY 4.0, Fonts: SIL OFL 1.1, Code: MIT License)
 */
.fa {
    font-family: var(--fa-style-family, "Font Awesome 6 Free");
    font-weight: var(--fa-style, 900);
}

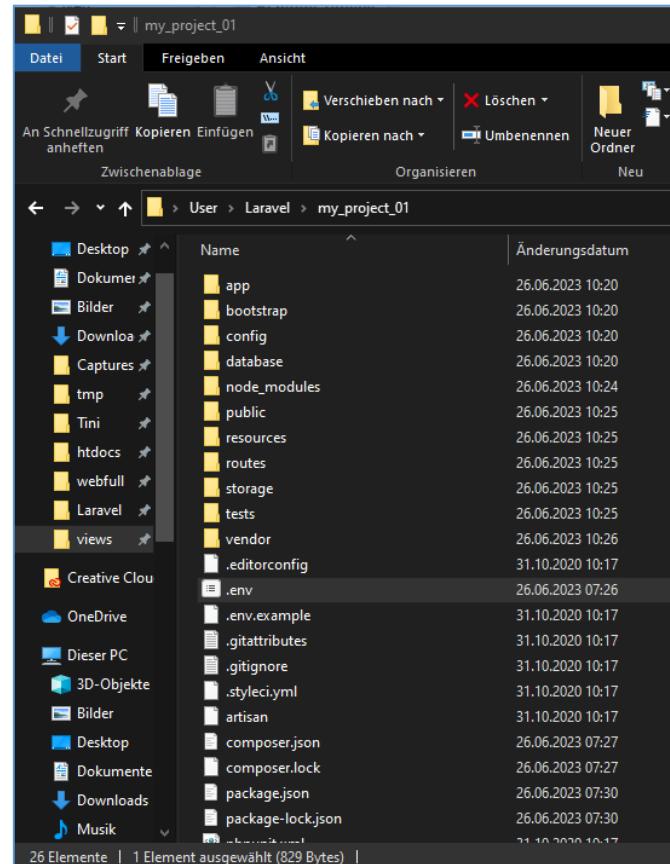
.fa,
.fa-classic,
```

Verbindung zur
Datenbank herstellen

1) XAMPP Starten und Datenbank anlegen



2) Verbindung zur Datenbank herstellen in der .env Datei



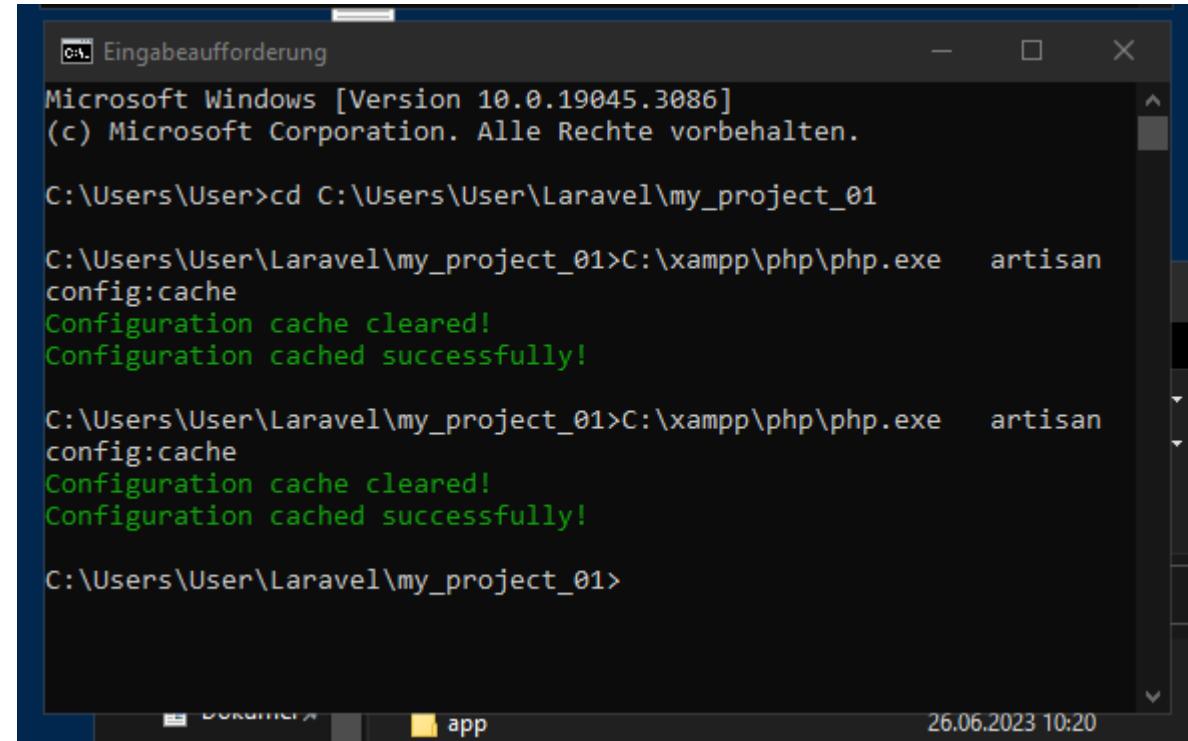
The screenshot shows the phpMyAdmin interface with the following details:

- Server: 127.0.0.1
- Datenbank: blog_laravel_webfull02
- Message: Es wurden keine Tabellen in der Datenbank gefunden.
- Form: Erzeuge Tabelle
- Fields: Name: [empty], Anzahl der Spalten: [empty]

```
C: > Users > User > Laravel > my_project_01 > .env
1 APP_NAME=Laravel
2 APP_ENV=local
3 APP_KEY=base64:o83F+IySdSRwscyZnAxNwzOushIy6loPpCSTzHwYNpU=
4 APP_DEBUG=true
5 APP_URL=http://localhost
6
7 LOG_CHANNEL=stack
8
9 DB_CONNECTION=mysql
10 DB_HOST=127.0.0.1
11 DB_PORT=3306
12 DB_DATABASE=blog_laravel_webfull02
13 DB_USERNAME=root
14 DB_PASSWORD=
15
16 BROADCAST_DRIVER=log
17 CACHE_DRIVER=file
18 QUEUE_CONNECTION=sync
19 SESSION_DRIVER=file
20 SESSION_LIFETIME=120
21
22 REDIS_HOST=127.0.0.1
23 REDIS_PASSWORD=null
24 REDIS_PORT=6379
25
26 MAIL_MAILER=smtp
27 MAIL_HOST=smtp.mailtrap.io
28 MAIL_PORT=2525
29 MAIL_USERNAME=null
30 MAIL_PASSWORD=null
31 MAIL_ENCRYPTION=null
32 MAIL_FROM_ADDRESS=null
33 MAIL_FROM_NAME="${APP_NAME}"
34
```

3) Ganz wichtig - jetzt Cache löschen per CMD

C:\xampp\php\php.exe artisan config:cache



```
Eingabeaufforderung
Microsoft Windows [Version 10.0.19045.3086]
(c) Microsoft Corporation. Alle Rechte vorbehalten.

C:\Users\User>cd C:\Users\User\Laravel\my_project_01

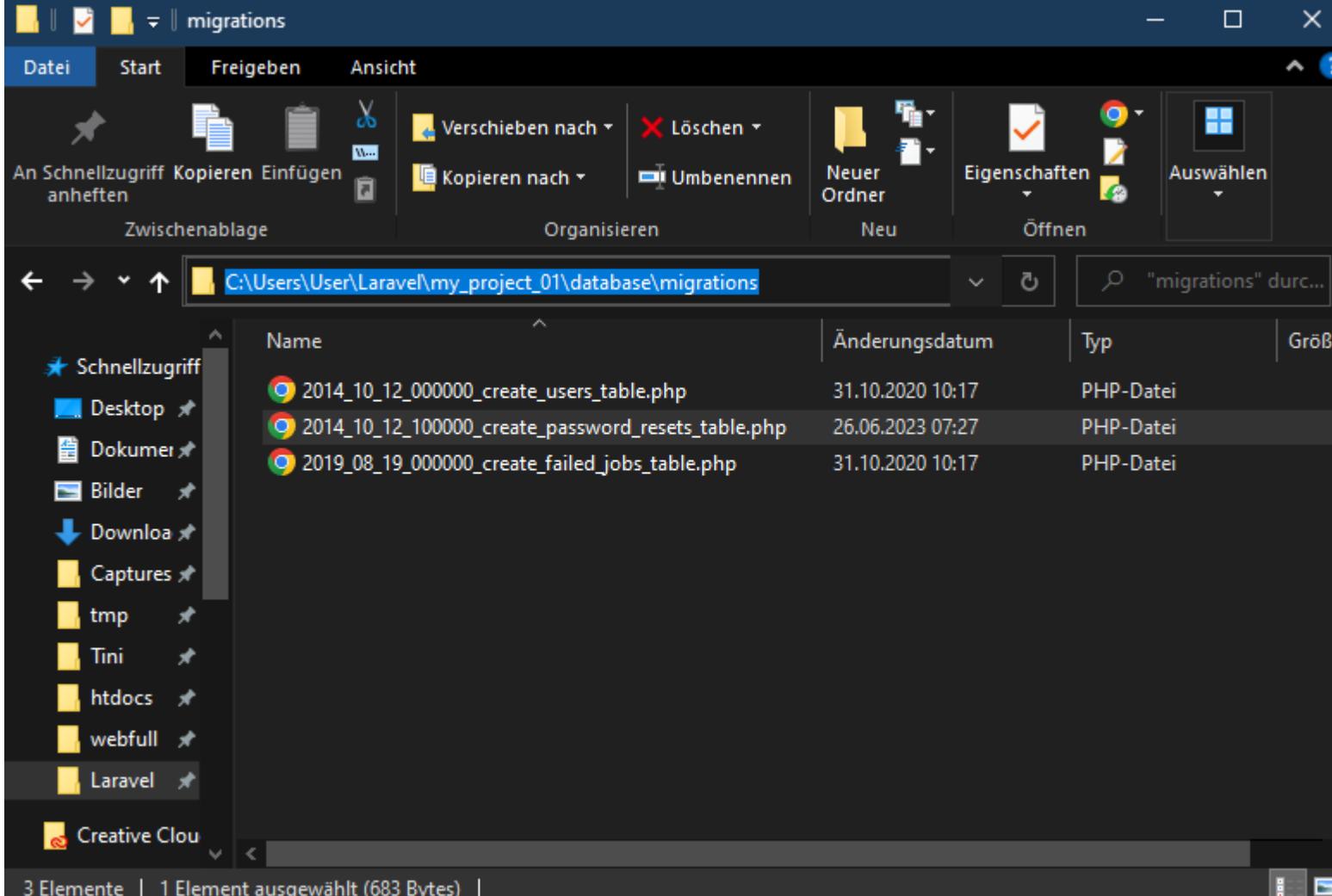
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan
config:cache
Configuration cache cleared!
Configuration cached successfully!

C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan
config:cache
Configuration cache cleared!
Configuration cached successfully!

C:\Users\User\Laravel\my_project_01>
```

Tabellen anlegen/bearbeiten

Login und Register automatisch schon vorhanden -> DBTabelle wird aut. wie angegeben angelegt



```
2014_10_12_000000_create_users_table.php - htdocs - Visual Studio Code - 2024-01-16 10:27:45

C:\Users\User\ Laravel\my_project_01\database\migrations\2014_10_12_000000_create_users_table.php

1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
8 {
9
10     /**
11      * Run the migrations.
12      *
13      * @return void
14      */
15     public function up()
16     {
17         Schema::create('users', function (Blueprint $table) {
18             $table->id();
19             $table->string('name');
20             $table->string('email')->unique();
21             $table->timestamp('email_verified_at')->nullable();
22             $table->string('password');
23             $table->rememberToken();
24             $table->timestamps();
25         });
26     }
27
28     /**
29      * Reverse the migrations.
30      *
31      * @return void
32      */
33     public function down()
34     {
35         Schema::dropIfExists('users');
36     }
37 }
```

1) Vorhandene Tabellen anlegen:

C:\xampp\php\php.exe artisan migrate

Die Tabellen werden automatisch erstellt

```
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62469 Closing
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62468 Closing
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62470 Accepted
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62470 [200]: GET /favicon.ico
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62470 Closing
[Tue Jun 27 08:09:49 2023] 127.0.0.1:62471 Accepted
[Tue Jun 27 08:11:20 2023] 127.0.0.1:62471 Closed without sending a request; it was probably just an unused speculative preconnection
[Tue Jun 27 08:11:20 2023] 127.0.0.1:62471 Closing

Eingabeaufforderung - C:\xampp\php\php.exe artisan serve
(c) Microsoft Corporation. Alle Rechte vorbehalten.
C:\Users\User>cd C:\Users\User\Laravel\my_project_01
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan config:cache
Configuration cache cleared!
Configuration cached successfully!
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.05 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.03 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.01 seconds)

C:\Users\User\Laravel\my_project_01>
```

The screenshot shows a Windows File Explorer window with the path "database > migrations". It lists several PHP files representing database migrations:

- 2014_10_12_000000_create_users_table.php (31.10.2020 10:17)
- 2014_10_12_100000_create_password_resets_table.php (26.06.2023 07:27)
- 2019_08_19_000000_create_failed_jobs_table.php (31.10.2020 10:17)

The screenshot shows the phpMyAdmin interface connected to a MySQL server at 127.0.0.1. The current database is "blog_laravel_webfull02". The left sidebar shows the database structure with tables: failed_jobs, migrations, password_resets, and users. The main panel displays the "Tabelle" (Table) view for these four tables. The table structure includes columns for Aktion (Action), Datensätze (Rows), Typ (Type), Kollation (Collation), Größe (Size), and Überhang (InnoDB only). The "password_resets" table has 3 rows and a size of 32,0 KiB. The "users" table has 0 rows and a size of 32,0 KiB.

Tabelle	Aktion	Datensätze	Typ	Kollation	Größe	Überhang
failed_jobs	Anzeigen Struktur Suche Einfügen Leeren Löschen	0	InnoDB	utf8mb4_unicode_ci	16,0 KiB	-
migrations	Anzeigen Struktur Suche Einfügen Leeren Löschen	3	InnoDB	utf8mb4_unicode_ci	16,0 KiB	-
password_resets	Anzeigen Struktur Suche Einfügen Leeren Löschen	0	InnoDB	utf8mb4_unicode_ci	32,0 KiB	-
users	Anzeigen Struktur Suche Einfügen Leeren Löschen	0	InnoDB	utf8mb4_unicode_ci	32,0 KiB	-

1a) DB users – Registration funktioniert

The image displays four screenshots illustrating the successful registration of a user in a Laravel application.

Top Left: Screenshot of the phpMyAdmin interface showing the structure of the 'users' table in the 'blog_laravel_webfull02' database. The table has columns: id (bigint(20), unsigned, primary key, auto-increment), name (varchar(255)), email (varchar(255)), email_verified_at (timestamp), password (varchar(255)), remember_token (varchar(100)), created_at (timestamp), and updated_at (timestamp).

Top Right: Screenshot of the 'Register' form from the 'MyProjekt' application. The form fields are: Name (Christin Lippold), E-Mail Address (crowtini@web.de), Password (*****), and Confirm Password (*****). A 'Register' button is at the bottom.

Bottom Left: Screenshot of the Windows File Explorer showing the project structure. It includes files like 'app.php' and 'User.php' under the 'app' folder.

Bottom Right: Screenshot of the 'Dashboard' page from the 'MyProjekt' application. The header shows 'Christin Lippold'. The dashboard area displays the message 'You are logged in!'. Below it is a table showing user data, with one row selected:

	id	name	email	email_verified_at	password	remember_token	created_at	updated_at
	1	Christin Lippold	crowtini@web.de	NULL	\$2y\$10\$3gUxw48oSU3Kj0CoaaJlOx7NRI8Z2ODih/GD88tP6...	NULL	2023-06-27 06:30:36	2023-06-27 06:30:36

2) Registrierung und Tabelle erweitern....

Tabellenspalten eintragen, dann migrieren und dann im Formular anlegen

The terminal window shows the following content:

```
logging.php 2014_10_12_000000_create_users_table.php register.blade.php
uebung01 > database > migrations > 2014_10_12_000000_create_users_table.php
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreateUsersTable extends Migration
{
8
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->id();
18             $table->string('name');
19             $table->string('email')->unique();
20             $table->timestamp('email_verified_at')->nullable();
21             $table->string('password');
22             $table->string('motto');
23             $table->string('ueber_mich');
24             $table->rememberToken();
25             $table->timestamps();
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      *
32      * @return void
33      */
34     public function down()
35     {
36         Schema::dropIfExists('users');
37     }
38 }
```

Output of the artisan migrate:fresh command:

```
C:\Users\User\Laravel\uebung01>C:\xampp\php\php.exe artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.03 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.02 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.01 seconds)
```

Below the terminal is a screenshot of a database table structure:

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra
1	id	bigint(20)		UNSIGNED	Nein	kein(e)		AUTO_INCREMENT
2	name	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)		
3	email	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)		
4	email_verified_at	timestamp			Ja	NULL		
5	password	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)		
6	motto	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)		
7	ueber_mich	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)		
8	remember_token	varchar(100)	utf8mb4_unicode_ci		Ja	NULL		
9	created_at	timestamp			Ja	NULL		
10	updated_at	timestamp			Ja	NULL		

The Visual Studio Code window shows the following blade template code:

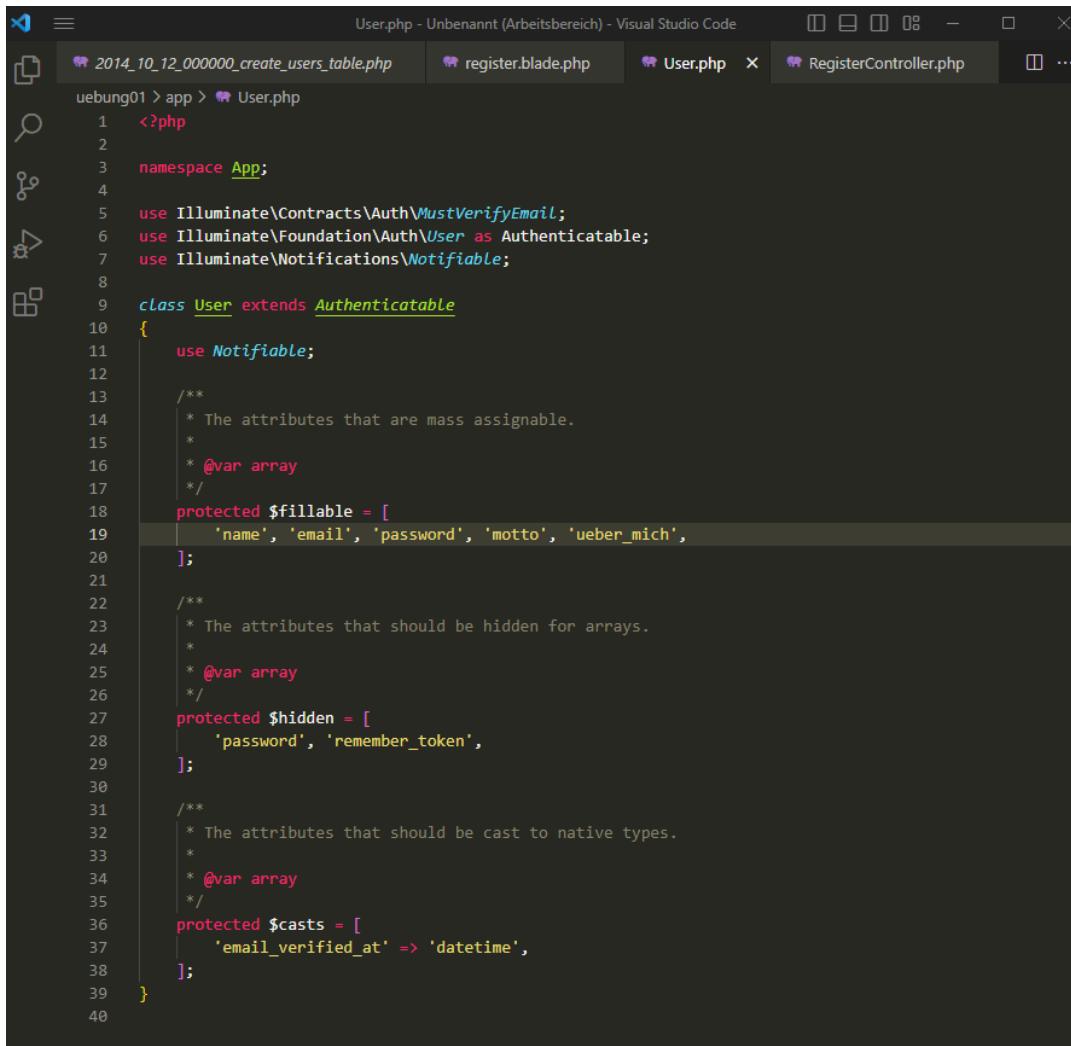
```
register.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code
register.blade.php register.blade.php User.php RegisterController.php
uebung01 > resources > views > auth > register.blade.php
1 <div class="form-group row">
2     <label for="password-confirm" class="col-md-4 col-form-label text-md-right">{{ __('Confirm Password') }}</label>
3
4     <div class="col-md-6">
5         <input id="password-confirm" type="password" class="form-control" name="password_confirmation" required autocomplete="new-password">
6     </div>
7
8     <div class="form-group row">
9         <label for="motto" class="col-md-4 col-form-label text-md-right">{{ __('Motto') }}</label>
10
11         <div class="col-md-6">
12             <input id="motto" type="text" class="form-control @error('motto') is-invalid @enderror" name="motto" value="{{ old('motto') }}" required autocomplete="motto" autofocus>
13
14             @error('motto')
15                 <span class="invalid-feedback" role="alert">
16                     <strong>{{ $message }}</strong>
17                 </span>
18             @enderror
19         </div>
20
21     <div class="form-group row">
22         <label for="ueber_mich" class="col-md-4 col-form-label text-md-right">{{ __('Ueber_Mich') }}</label>
23
24         <div class="col-md-6">
25             <input id="ueber_mich" type="text" class="form-control @error('ueber_mich') is-invalid @enderror" name="ueber_mich" value="{{ old('name') }}" required autocomplete="ueber_mich" autofocus>
26
27             @error('ueber_mich')
28                 <span class="invalid-feedback" role="alert">
29                     <strong>{{ $message }}</strong>
30                 </span>
31             @enderror
32         </div>
33     </div>
34
35     <div class="form-group row">
36         <button type="submit" class="btn btn-primary">Register
37     </div>
38 
```

The browser window shows a registration form titled "Register". The fields are:

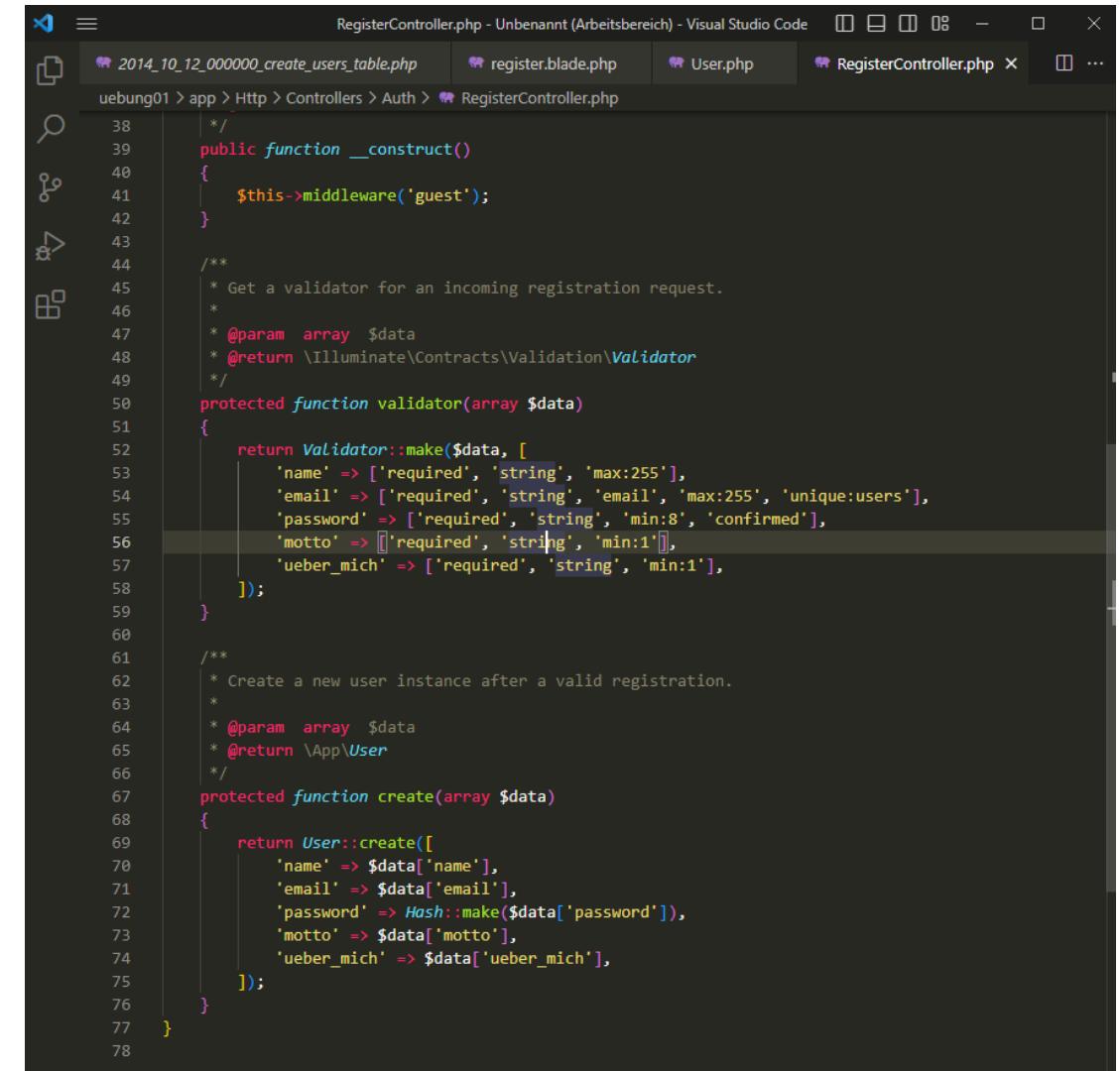
- Name
- E-Mail Address
- Password
- Confirm Password
- Motto
- Ueber_Mich

At the bottom right of the form is a blue "Register" button.

3) Registrierung erweitern... Übertragungsdaten festlegen



```
1 <?php
2
3 namespace App;
4
5 use Illuminate\Contracts\Auth\MustVerifyEmail;
6 use Illuminate\Foundation\Auth\User as Authenticatable;
7 use Illuminate\Notifications\Notifiable;
8
9 class User extends Authenticatable
{
    use Notifiable;
11
12    /**
13     * The attributes that are mass assignable.
14     *
15     * @var array
16     */
17 protected $fillable = [
18     'name', 'email', 'password', 'motto', 'ueber_mich',
19 ];
20
21 /**
22 * The attributes that should be hidden for arrays.
23 *
24 * @var array
25 */
26 protected $hidden = [
27     'password', 'remember_token',
28 ];
29
30 /**
31 * The attributes that should be cast to native types.
32 *
33 * @var array
34 */
35 protected $casts = [
36     'email_verified_at' => 'datetime',
37 ];
38 }
39 }
```



```
38 */
39 public function __construct()
40 {
41     $this->middleware('guest');
42 }
43
44 /**
45 * Get a validator for an incoming registration request.
46 *
47 * @param array $data
48 * @return \Illuminate\Contracts\Validation\Validator
49 */
50 protected function validator(array $data)
51 {
52     return Validator::make($data, [
53         'name' => ['required', 'string', 'max:255'],
54         'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],
55         'password' => ['required', 'string', 'min:8', 'confirmed'],
56         'motto' => ['required', 'string', 'min:1'],
57         'ueber_mich' => ['required', 'string', 'min:1'],
58     ]);
59 }
60
61 /**
62 * Create a new user instance after a valid registration.
63 *
64 * @param array $data
65 * @return \App\User
66 */
67 protected function create(array $data)
68 {
69     return User::create([
70         'name' => $data['name'],
71         'email' => $data['email'],
72         'password' => Hash::make($data['password']),
73         'motto' => $data['motto'],
74         'ueber_mich' => $data['ueber_mich'],
75     ]);
76 }
77 }
78 }
```

Hilfdatei für Abfragen:

C:\xampp\php\php.exe artisan make:model -h
(-h = Help)

```
C:\ Eingabeaufforderung
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan make:model -h
Description:
Create a new Eloquent model class

Usage:
make:model [options] [--] <name>

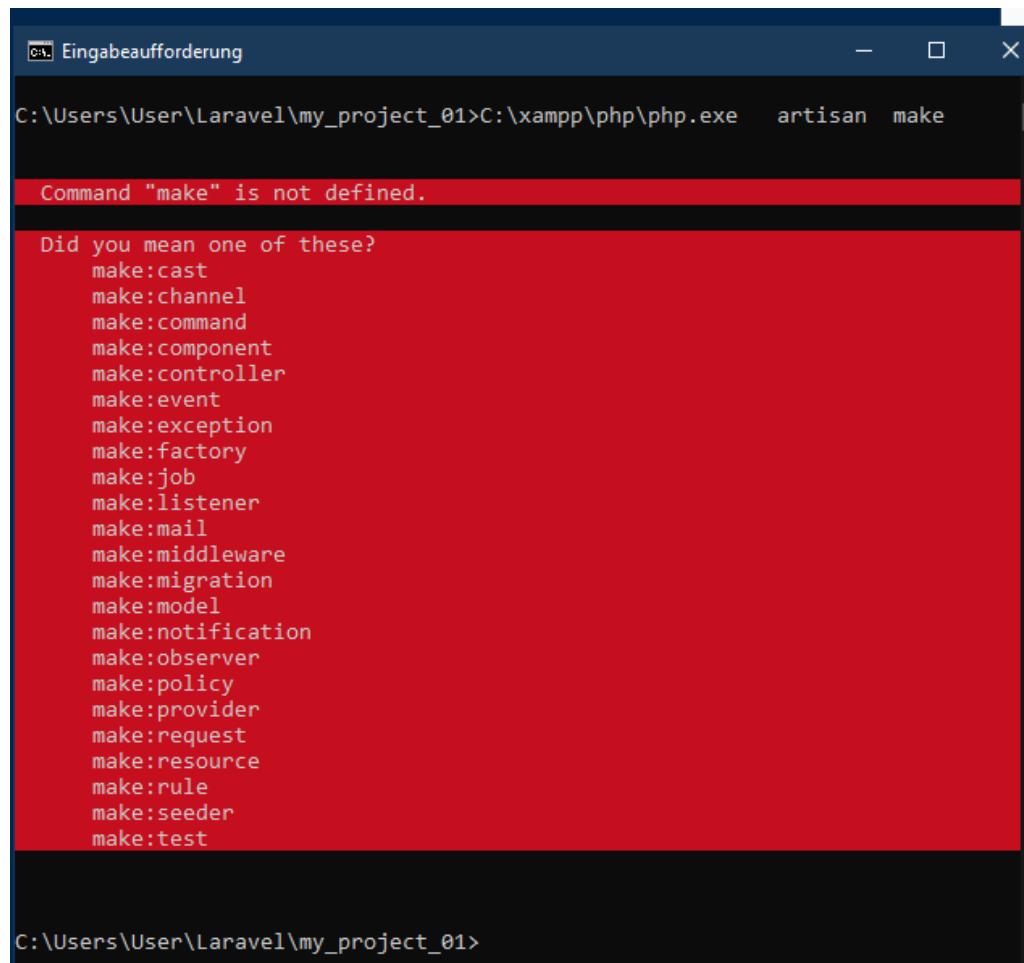
Arguments:
  name          The name of the class

Options:
  -a, --all      Generate a migration, seeder, factory, and resource controller for the model
  -c, --controller Create a new controller for the model
  -f, --factory   Create a new factory for the model
  --force         Create the class even if the model already exists
  -m, --migration Create a new migration file for the model
  -s, --seed       Create a new seeder file for the model
  -p, --pivot     Indicates if the generated model should be a custom intermediate table model
  -r, --resource   Indicates if the generated controller should be a resource controller
  --api           Indicates if the generated controller should be an API controller
  -h, --help       Display help for the given command. When no command is given display help for the list command
  -q, --quiet      Do not output any message
  -V, --version    Display this application version
  --ansi|--no-ansi Force (or disable --no-ansi) ANSI output
  -n, --no-interaction Do not ask any interactive question
  --env[=ENV]      The environment the command should run under
  -v|vv|vvv, --verbose Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

C:\Users\User\Laravel\my_project_01>
```

Make kann folgendes:

C:\xampp\php\php.exe artisan make



```
Eingabeaufforderung
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan make

Command "make" is not defined.

Did you mean one of these?
make:cast
make:channel
make:command
make:component
make:controller
make:event
make:exception
make:factory
make:job
make:listener
make:mail
make:middleware
make:migration
make:model
make:notification
make:observer
make:policy
make:provider
make:request
make:resource
make:rule
make:seeder
make:test

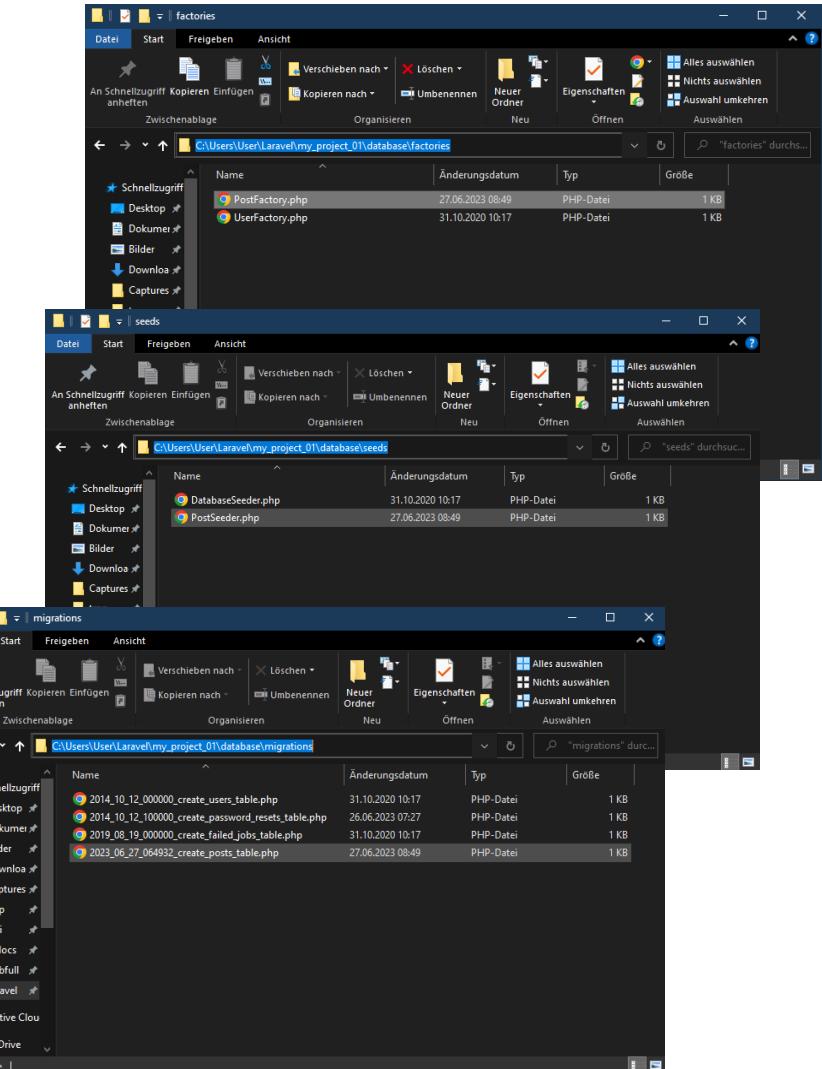
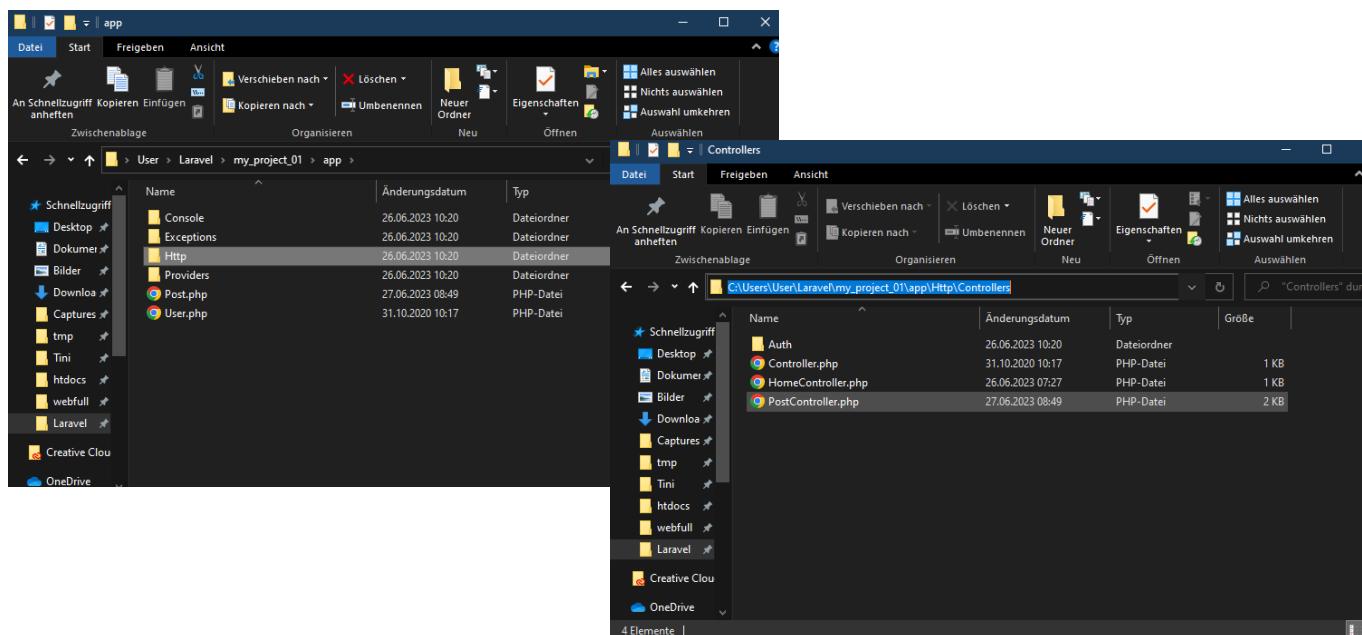
C:\Users\User\Laravel\my_project_01>
```

make:model steht für Dateianlage... alles wird auto. dort erstellt wo es stehen muss

```
Eingabeaufforderung

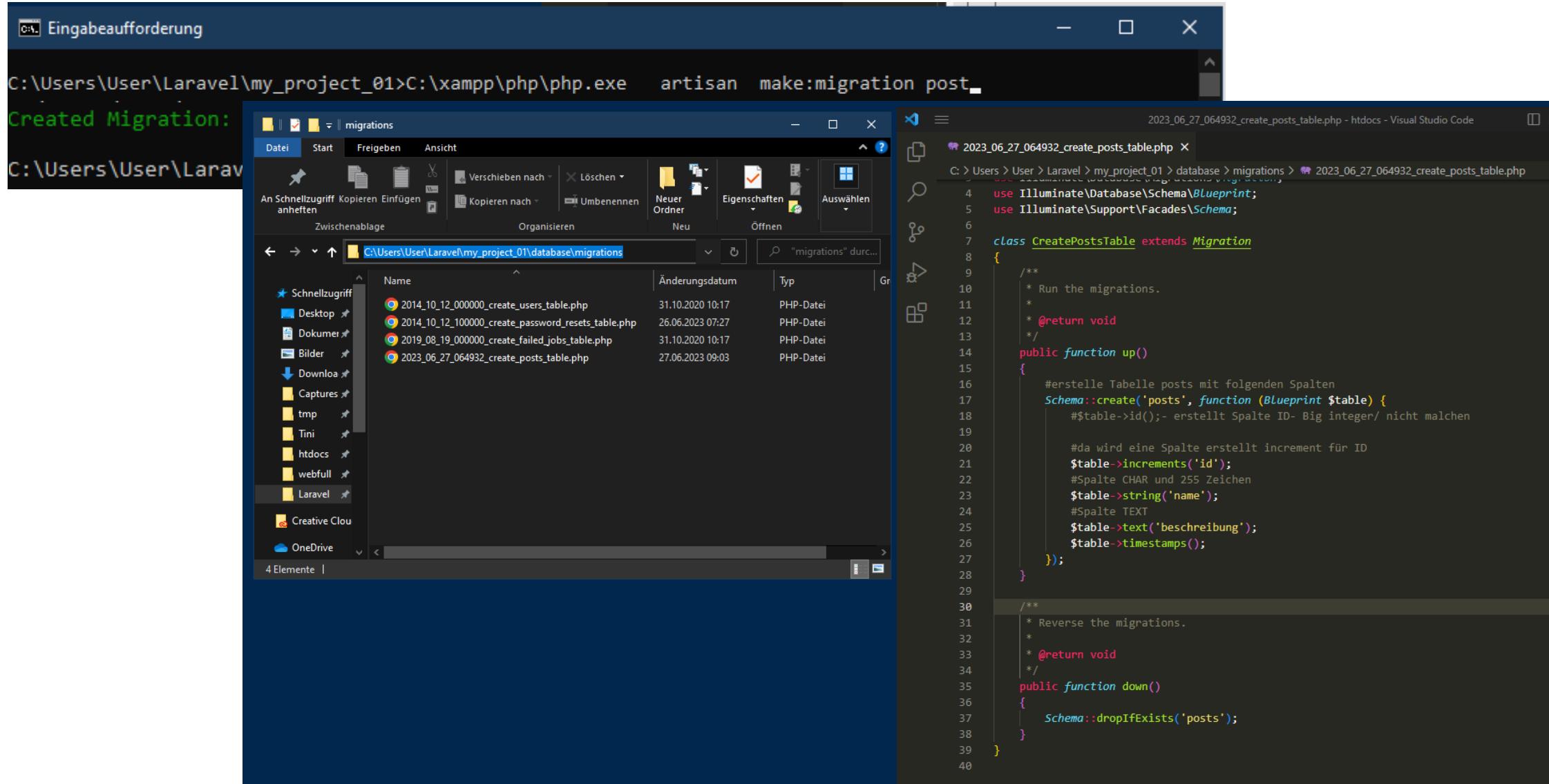
C:\Users\User\ Laravel\my_project_01>C:\xampp\php\php.exe artisan make:model Post -a
Model created successfully.
Factory created successfully.
Created Migration: 2023_06_27_064932_create_posts_table
Seeder created successfully.
Controller created successfully.

C:\Users\User\ Laravel\my_project_01>
```



1) Eigene Tabelle erstellen – Spalten festlegen

C:\xampp\php\php.exe artisan make:migration post



Eingabeaufforderung

C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan make:migration post

Created Migration:

C:\Users\User\Laravel\my_project_01>

Name	Änderungsdatum	Typ
2014_10_12_000000_create_users_table.php	31.10.2020 10:17	PHP-Datei
2014_10_12_100000_create_password_resets_table.php	26.06.2023 07:27	PHP-Datei
2019_08_19_000000_create_failed_jobs_table.php	31.10.2020 10:17	PHP-Datei
2023_06_27_064932_create_posts_table.php	27.06.2023 09:03	PHP-Datei

```
2023_06_27_064932_create_posts_table.php - htdocs - Visual Studio Code
C:\Users\User\Laravel\my_project_01>
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class CreatePostsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('posts', function (Blueprint $table) {
17             $table->id(); - erstellt Spalte ID- Big integer/ nicht malchen
18
19             #da wird eine Spalte erstellt increment für ID
20             $table->increments('id');
21             #Spalte CHAR und 255 Zeichen
22             $table->string('name');
23             #Spalte TEXT
24             $table->text('beschreibung');
25             $table->timestamaps();
26         });
27     }
28
29     /**
30      * Reverse the migrations.
31      *
32      * @return void
33      */
34     public function down()
35     {
36         Schema::dropIfExists('posts');
37     }
38 }
```

2) Eigene Tabelle erstellen – in DB übertrage

C:\xampp\php\php.exe artisan migrate

```
C:\Users\User\ Laravel\my_project_01>C:\xampp\php\php.exe artisan migrate
Migrating: 2023_06_27_064932_create_posts_table
Migrated: 2023_06_27_064932_create_posts_table (0.02 seconds)

C:\Users\User\ Laravel\my_project_01>
```

The screenshot shows the phpMyAdmin interface for a database named "blog_laravel_webfull02". The left sidebar displays the database structure with tables: failed_jobs, migrations, password_resets, posts, and users. The right panel shows a list of all tables in the database, including their names, actions (Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen), and details like Type, Collation, and Size.

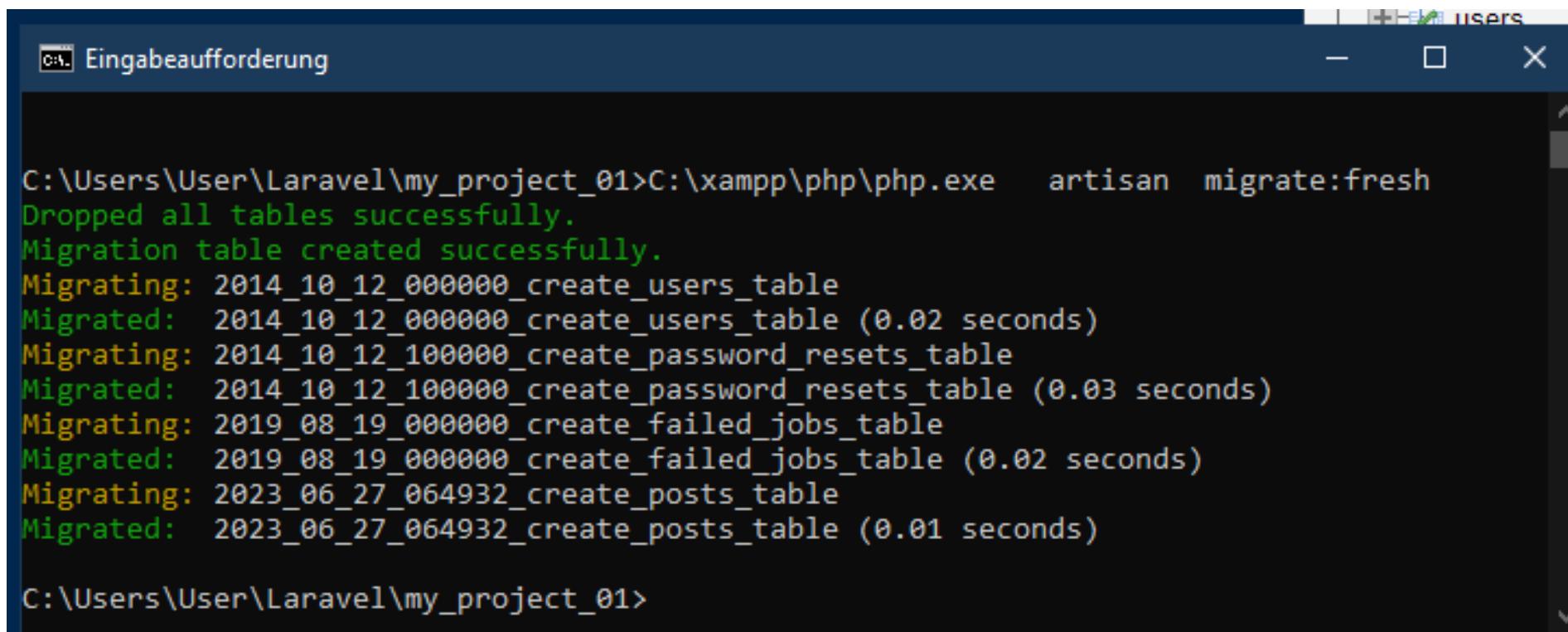
Tabelle	Aktion	Datensätze	Type	Kollation	Größe
failed_jobs	Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen	0	InnoDB	utf8mb4_unicode_ci	16,0 Kib
migrations	Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen	4	InnoDB	utf8mb4_unicode_ci	16,0 Kib
password_resets	Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen	0	InnoDB	utf8mb4_unicode_ci	32,0 Kib
posts	Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen	0	InnoDB	utf8mb4_unicode_ci	16,0 Kib
users	Anzeigen, Struktur, Suche, Einfügen, Leeren, Löschen	1	InnoDB	utf8mb4_unicode_ci	32,0 Kib
5 Tabellen	Gesamt	5	InnoDB	utf8mb4_general_ci	112,0 Kib

The screenshot shows the phpMyAdmin interface for the "posts" table within the "blog_laravel_webfull02" database. The left sidebar shows the database structure. The right panel displays the "Tabellenstruktur" (Table Structure) for the "posts" table, listing columns: id, name, beschreibung, created_at, and updated_at. Each column has its data type, collation, attributes (Null, Standard, Kommentare, Extra), and actions (Bearbeiten, Löschen, Mehr).

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
1	id	int(10)	UNSIGNED	Nein	kein(e)			AUTO_INCREMENT	Bearbeiten, Löschen, Mehr
2	name	varchar(255)	utf8mb4_unicode_ci	Nein	kein(e)				Bearbeiten, Löschen, Mehr
3	beschreibung	text	utf8mb4_unicode_ci	Nein	kein(e)				Bearbeiten, Löschen, Mehr
4	created_at	timestamp		Ja	NULL				Bearbeiten, Löschen, Mehr
5	updated_at	timestamp		Ja	NULL				Bearbeiten, Löschen, Mehr

3) DB wird aktualisiert... Spalten hinzugefügt und co. ... aber auch alle Inhalte gelöscht

C:\xampp\php\php.exe artisan migrate:fresh



```
C:\Users\User\ Laravel\my_project_01>C:\xampp\php\php.exe artisan migrate:fresh
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (0.02 seconds)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (0.03 seconds)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (0.02 seconds)
Migrating: 2023_06_27_064932_create_posts_table
Migrated: 2023_06_27_064932_create_posts_table (0.01 seconds)

C:\Users\User\ Laravel\my_project_01>
```

3a) Spalte in vorhanden Tabelle einfügen ...

Neu anlegen + Verbindung eintragen + migration

C:\xampp\php\php.exe artisan make:migration user_id_in_posts_table

```
C:\Users\User\ Laravel\my_project_01>C:\xampp\php\php.exe artisan make:migration user_id_in_posts_table
Created Migration: 2023_06_29_115549_user_id_in_posts_table

C:\Users\User\ Laravel\my_project_01>
```

migrations

Datei Start Freigeben Ansicht

An Schnellzugriff Kopieren Einfügen anhängen Neuer Ordner Neu Eigenschaften

Zwischenablage Organisieren Öffnen

2023_06_29_115222_user_id.php 2023_06_29_115549_user_id_in_posts_table.php X 2014_10_12_000000_create_users_table.php

my_project_01 > database > migrations > 2023_06_29_115549_user_id_in_posts_table.php

```
1 <?php
2
3 use Illuminate\Database\Migrations\Migration;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Support\Facades\Schema;
6
7 class UserIdInPostsTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::table('posts', function (Blueprint $table) {
17
18             //erstelle eine Spalte die
19             //Unsigned + BigInteger + nicht Null ist und user_id heißt
20             //wird hinter Spalte Id eingefügt
21             $table->unsignedBigInteger('user_id')->nullable()->after('id');
22
23
24             //user_id soll foreignschlüssel werden
25             //und mit der Spalte Id von users in Verbindung treten
26             //wenn User gelöscht wird sollen die Posts auch gelöscht werden
27             $table->foreign('user_id')
28                 ->references('id')->on('users')
29                 ->onDelete('cascade');
30         });
31     }
32
33 }
```

C:\xampp\php\php.exe artisan migrate

C:\Users\User\ Laravel\my_project_01>C:\xampp\php\php.exe artisan migrate

Migrating: 2023_06_29_115222_user_id
Migrated: 2023_06_29_115222_user_id (0.01 seconds)
Migrating: 2023_06_29_115549_user_id_in_posts_table
Migrated: 2023_06_29_115549_user_id_in_posts_table (0.09 seconds)

Screenshot of the MySQL Workbench interface showing the 'posts' table structure and constraints.

The 'posts' table structure:

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
1	id	int(10)		UNSIGNED	Nein	kein(e)		AUTO_INCREMENT	Be
2	user_id	bigint(20)		UNSIGNED	Ja	NULL			Be
3	name	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)			Be
4	beschreibung	text	utf8mb4_unicode_ci		Nein	kein(e)			Be
5	created_at	timestamp			Ja	NULL			Be

The 'posts_user_id_foreign' constraint configuration:

Aktionen	Constrainteigenschaften	Spalte	Beschränkung für auswärtige Schlüssel (INNODB)			
			Datenbank	Tabelle	Spalte	
Löschen	ON DELETE CASCADE	user_id	blog_laravel_we	users	id	
	ON UPDATE RESTRICT	+ Spalte hinzufügen				
	Constraint-Name					
	ON DELETE RESTRICT		blog_laravel_we			
	ON UPDATE RESTRICT	+ Spalte hinzufügen				
	+ Constraint hinzufügen					

Fehlermeldungen

Fehlermeldungen in Laravel

C:\Users\User\ Laravel\my_project_01\

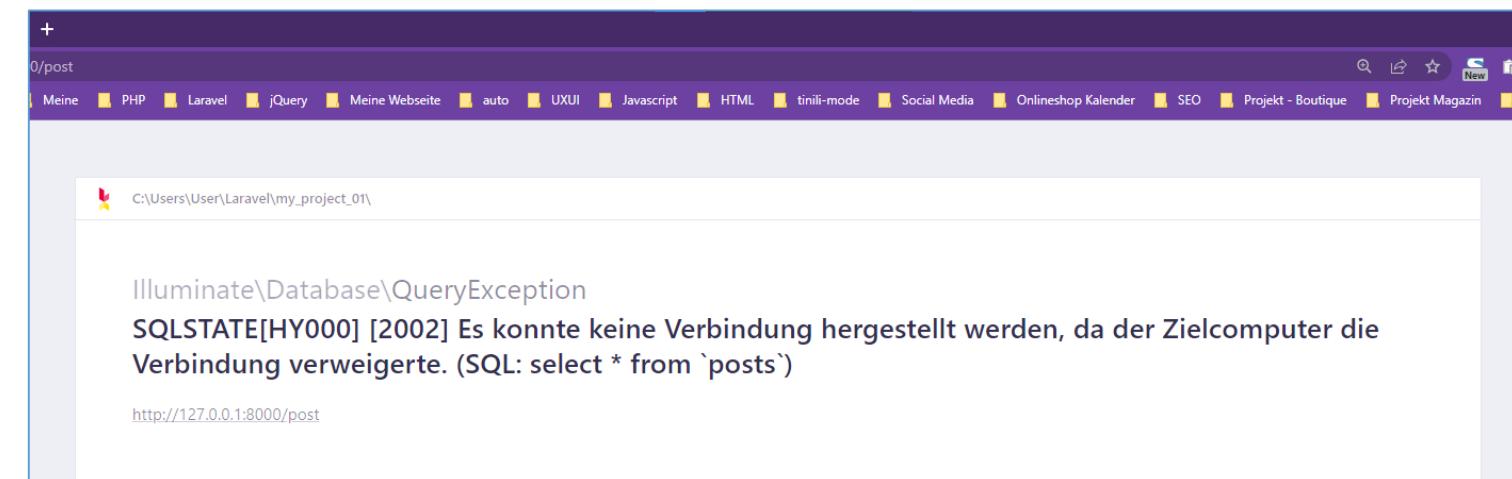
ErrorException
Undefined variable: daten (View:
C:\Users\User\ Laravel\my_project_01\resources\views\referenzen.blade.php)
<http://127.0.0.1:8000/referenzen>

\$daten is undefined

Hide solutions

	Stack trace	Request	App	User	Context	Debug	Share
1	Illuminate\Foundation\Bootstrap\HandleExceptions::handleError C:\Users\User\my_project_01\resources\views\referenzen.blade.php:15						
2			@extends('layouts.app')				
3			@section('content')				
4			<div class="container">				
5			<div class="row">				
6			<div class="col-md-12">				
7			<div class="card">				
8			<div class="card-header">				
9			Meine Referenzen				
10			</div>				
11			<div class="card-body">				
12			<p>				
13			Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua.				
14			</p>				
15			@foreach(\$daten AS \$user)				
16			<ul class="list-group m-2">				
17			<li class="list-group-item">Name: {{ \$user['name'] }}				
18			<li class="list-group-item">Vorname: {{ \$user['vorname'] }}				
19			<li class="list-group-item">Wohnort: {{ \$user['ort'] }}				
20							
21			@endforeach				
22			</div>				
23			</div>				
24			</div>				
25			</div>				
26			</div>				
27			@endsection				
28							

Fehlermeldungen In Laravel kein XAMPP gestartet



The screenshot shows a browser window with a purple header bar containing various navigation links. Below the header is a main content area with a light gray background. At the top of the content area, there is a small icon followed by the path 'C:\Users\User\Laravel\my_project_01\'. Below this, the error message is displayed in two lines: 'Illuminate\Database\QueryException' and 'SQLSTATE[HY000] [2002] Es konnte keine Verbindung hergestellt werden, da der Zielcomputer die Verbindung verweigerte. (SQL: select * from `posts`)'.

Below the error message, the URL 'http://127.0.0.1:8000/post' is shown. At the bottom of the page, there is a dark footer bar with several tabs: 'Stack trace', 'Request', 'App', 'User', 'Context', 'Debug', and 'Share'. The 'Stack trace' tab is currently selected, showing a detailed call stack with line numbers and file paths. The code snippets in the stack trace are as follows:

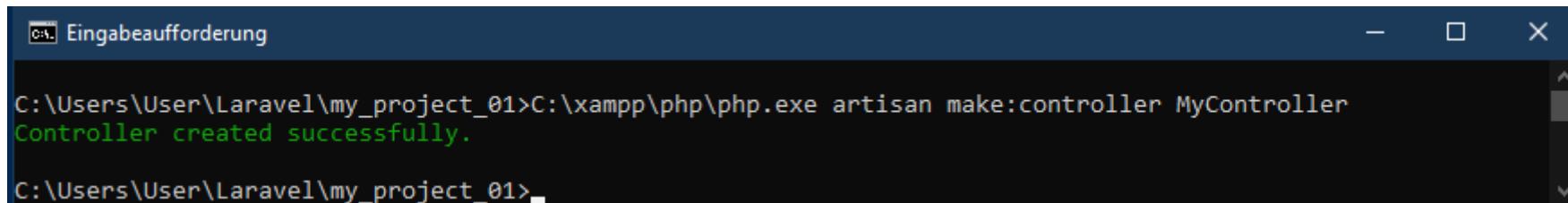
```
Illuminate\Database\Connection::runQueryCallback
C:\Users\User\Laravel\my_project_01\vendor\laravel\framework\src\Illuminate\Database\Connection.php:671
54 Illuminate\Database\Connection 656     * @throws \Illuminate\Database\QueryException
53 Illuminate\Database\Connection 657     */
52 Illuminate\Database\Connection 658     protected function runQueryCallback($query, $bindings, Closure $callback)
659     {
660         // To execute the statement, we'll simply call the callback, which will actually
661         // run the SQL against the PDO connection. Then we can calculate the time it
662         // took to execute and log the query SQL, bindings and time in our memory.
663         try {
664             $result = $callback($query, $bindings);
665         }
666
667         // If an exception occurs when attempting to run a query, we'll format the error
668         // message to include the bindings with SQL, which will make this exception a
669         // lot more helpful to the developer instead of just the database's errors.
670     catch (\Exception $e) {
```

Controller

Controller anlegen:

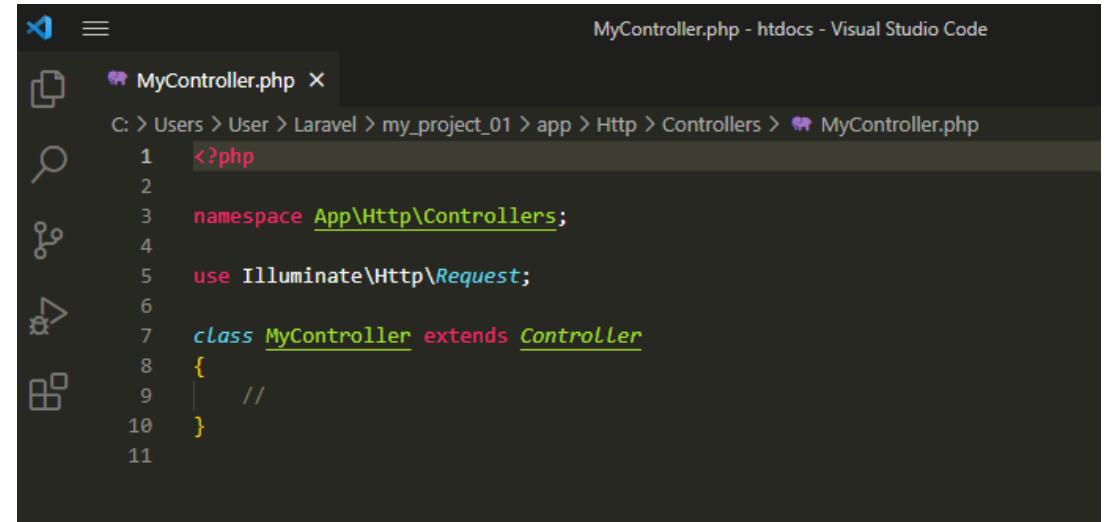
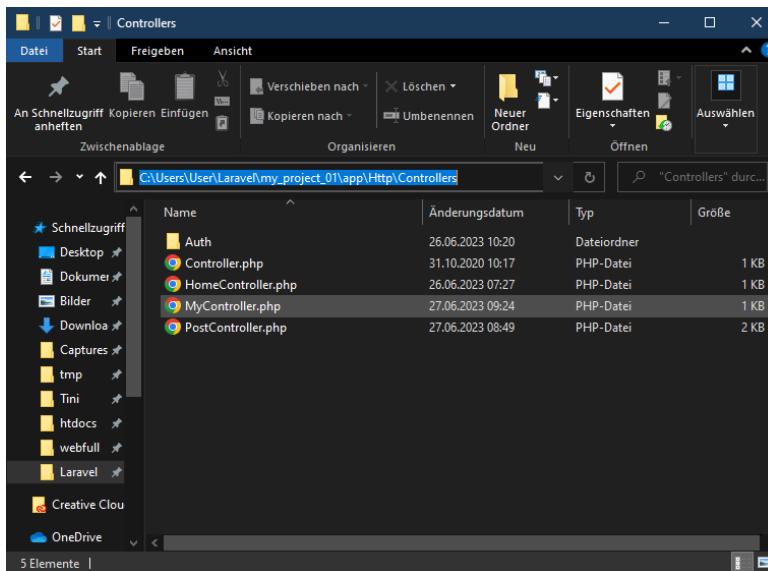
Controller spielen in Laravel eine sehr große Rolle

C:\xampp\php\php.exe artisan make:controller MyControllerBezeichnung



```
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan make:controller MyController
Controller created successfully.

C:\Users\User\Laravel\my_project_01>
```

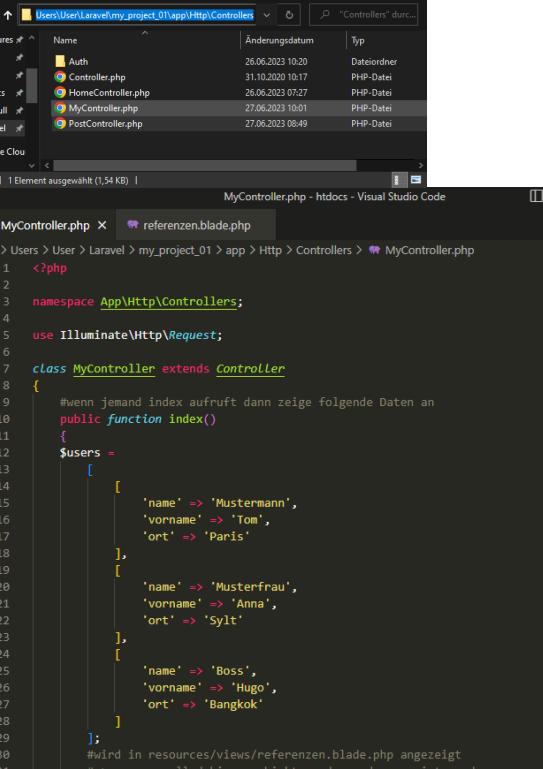


The screenshot shows the Visual Studio Code interface with the title 'MyController.php - htdocs - Visual Studio Code'. The code editor displays the following PHP code:

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class MyController extends Controller
8 {
9     //
10 }
```

Daten aus einem Array auslesen und anzeigen

Hier schreibt man Funktionen und Daten und/oder greift auf DB zu



The screenshot shows a Visual Studio Code window with the following details:

- File Explorer:** Shows a file tree for a Laravel project at `Users\User\ Laravel\my_project_01\app\Http\Controllers`. The tree includes `Captures`, `tmp`, `Tini`, `htdocs`, `webfull`, and `Laravel`. A file named `MyController.php` is selected.
- Search Bar:** Displays the search term `"Controllers" durch...`.
- Code Editor:** The file `MyController.php` is open. The code defines a controller class `MyController` extending `Controller`. It contains a method `index()` that returns a view with data from an array `$users`. The array contains three users with their names, surnames, and cities.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 class MyController extends Controller
8 {
9     #wenn jemand index aufruft dann zeige folgende Daten an
10    public function index()
11    {
12        $users =
13        [
14            [
15                'name' => 'Mustermann',
16                'vorname' => 'Tom',
17                'ort' => 'Paris'
18            ],
19            [
20                'name' => 'Musterfrau',
21                'vorname' => 'Anna',
22                'ort' => 'Sylt'
23            ],
24            [
25                'name' => 'Boss',
26                'vorname' => 'Hugo',
27                'ort' => 'Bangkok'
28            ]
29        ];
30        #wird in resources\views\referenzen.blade.php angezeigt
31        #->... was soll dahin geschickt werden und angezeigt werden
32        #with ('wei soll ich die ansprechen', welche daten soll der Inhalt sein)
33        return view('referenzen')->with('daten',$users);
34    }
35}
```
- Bottom Status Bar:** Shows "5 Elemente | 1 Element ausgewählt (1,54 KB)" and the file name "MyController.php - htdocs - Visual Studio Code".

The screenshot shows a Windows desktop environment with several open windows. In the foreground, the Visual Studio Code editor is open, displaying a PHP blade template named 'referenzen.blade.php'. The code uses the Laravel Blade templating engine. It includes sections for 'content' and 'referenzen'. The 'referenzen' section contains a list group with items for Name, Vorname, and Wohnort, each associated with a user object. The template also includes placeholder text (Lorem ipsum) and various CSS classes like 'container', 'row', 'col-md-12', 'card', 'card-header', and 'list-group-item'. The background shows a file explorer window with files like 'kontakt.blade.php' and 'referenzen.blade.php' listed. The taskbar at the bottom has icons for Start, Task View, File Explorer, and a search bar.

```
C:\> Users > User > Laravel > my_project_01 > resources > views > referenzen.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-12">
7       <div class="card">
8         <div class="card-header">
9           Meine Referenzen
10        </div>
11        <div class="card-body">
12          <p>
13            Lorem ipsum dolor sit amet, consetetur sadipscing elitr,  

14            sed diam nonumy eirmod tempor invidunt ut labore et  

15            dolore magna aliquyam erat, sed diam voluptua. At vero  

16            eos et accusam et justo duo dolores et ea rebum. Stet  

17            clita kasd gubergren, no sea takimata sanctus est Lorem  

18            ipsum dolor sit amet. Lorem ipsum dolor sit amet,  

19            consetetur sadipscing elitr, sed diam nonumy eirmod  

20            tempor invidunt ut labore et dolore magna aliquyam erat,  

21            sed diam voluptua. At vero eos et accusam et justo duo  

22            dolores et ea rebum. Stet clita kasd gubergren, no sea  

23            takimata sanctus est Lorem ipsum dolor sit amet. Lorem  

24            ipsum dolor sit amet, consetetur sadipscing elitr, sed  

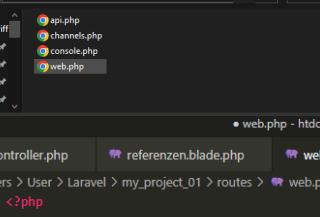
25            diam nonumy eirmod tempor invidunt ut labore et dolore  

26            magna aliquyam erat, sed diam voluptua. At vero eos et  

27            accusam et justo duo dolores et ea rebum. Stet clita kasd  

28            gubergren, no sea takimata sanctus est Lorem ipsum dolor  

29            sit amet.
30        </p>
31        @foreach($daten AS $user)
32          <ul class="list-group m-2">
33            <li class="list-group-item">Name: {{ $user['name'] }}</li>
34            <li class="list-group-item">Vorname: {{ $user['vorname'] }}</li>
35            <li class="list-group-item">Wohnort: {{ $user['ort'] }}</li>
36          </ul>
37        @endforeach
38      </div>
39    </div>
40  </div>
41 @endsection
```



MyProjekt Startseite Über uns Meine Referenzen Kontakt Login Register

Meine Referenzen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

Name: Mustermann
Vorname: Tom
Wohnort: Paris
Name: Musterfrau
Vorname: Anna
Wohnort: Sylt
Name: Boss
Vorname: Hugo
Wohnort: Bangkok

```
routes/api.php
routes/web.php

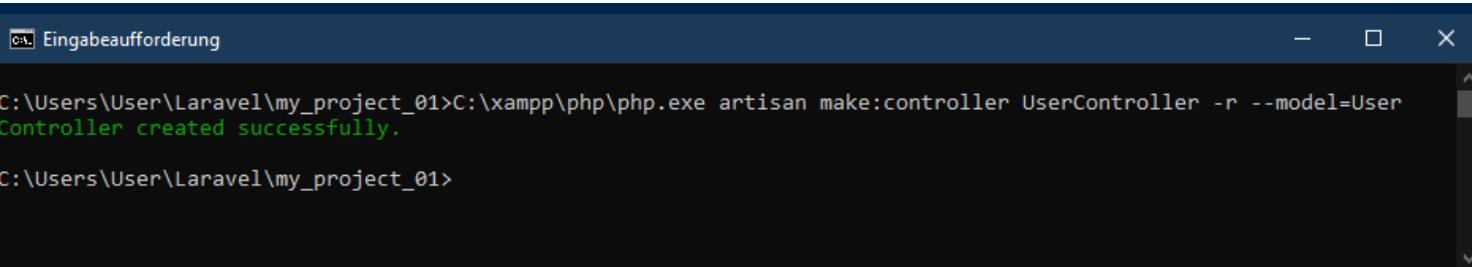
C:\Users\user\Laravel\my_project_01\routes> web.php - htdocs

MyController.php referenzen.blade.php web.php

1 <?php
2
3 use Illuminate\Support\Facades\Route;
4
5 /*
6 |-----|
7 | Web Routes
8 |-----|
9
10 | Here is where you can register web routes.
11 | routes are loaded by the RouteServiceProvider
12 | contains the "web" middleware group. Now d
13 |
14 */
15 #holt die Daten mit $_GET (post = $_POST)
16 #namenlose Funktion wird ausgeführt
17 #/gruss - fals in der Adresszeile gruss ste
18 Route::get('/gruss', function() {
19     echo "Hallo Welt, wie geht es dir?";
20     return "Guten Tag";
21 });
22
23 # / - steht für die Startseite
24 Route::get('/', function() {
25     return view('startseite');
26 });
27 Route::get('/about', function() {
28     return view('about');
29 });
30 Route::get('/referenzen', function() {
31     return view('referenzen');
32 });
33
34 #sucht wenn referenzen in Adresszeile nach Datei mit MyController
35 #und führt die Funktion index aus
36 Route::get('/referenzen', 'MyController@index');
37
38
39 Route::get('/adresse', function() {
40     return view('kontakt');
41 });
42
43 Auth::routes();
44
45 Route::get('/home', 'HomeController@index')->name('home');
```

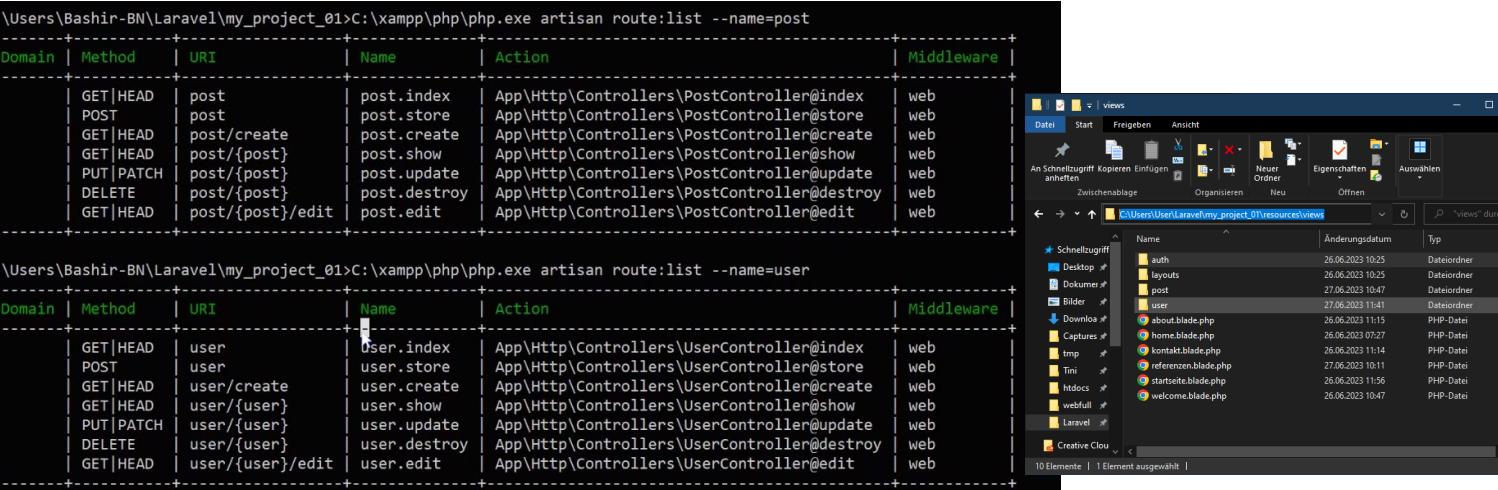
Controller regelt den Umgang mit Befehlen wie die Verbindung zu User erstellt wird

C:\xampp\php\php.exe artisan make:controller UserController -r --model=User

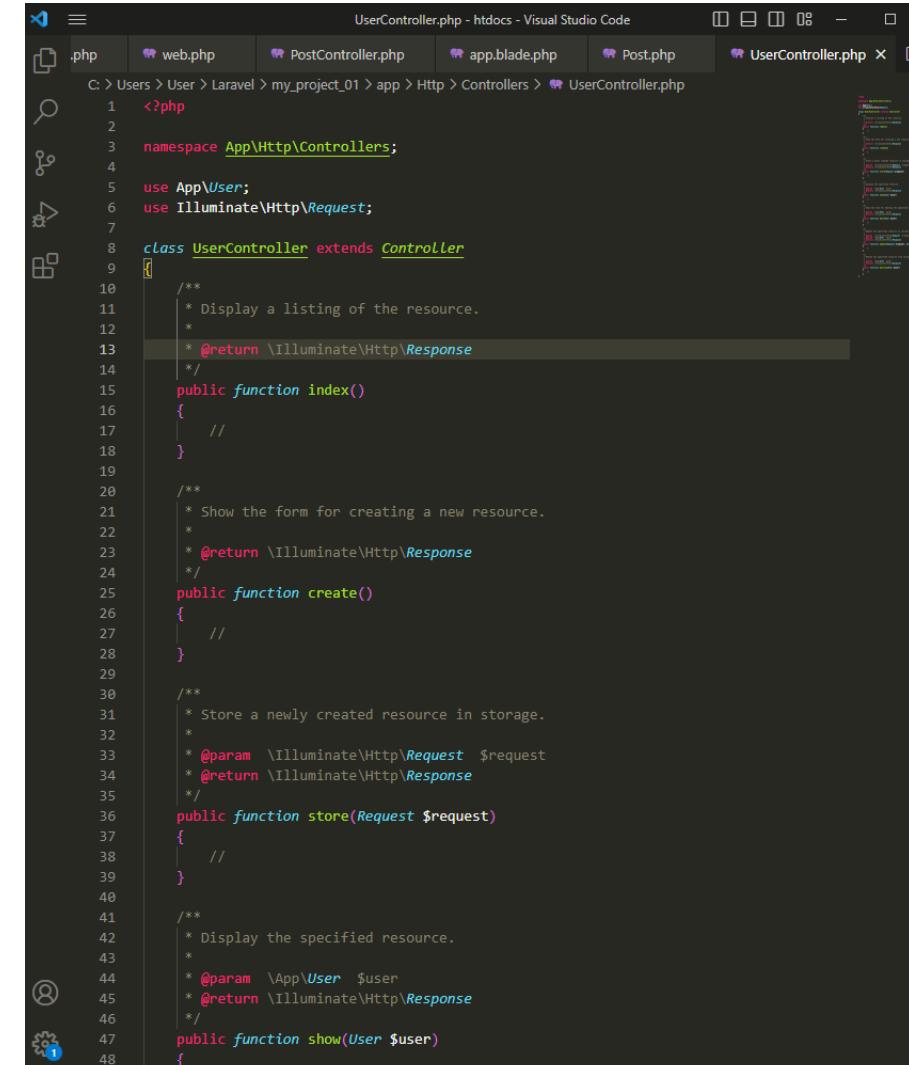


```
Eingabeaufforderung
C:\Users\User\Laravel\my_project_01>C:\xampp\php\php.exe artisan make:controller UserController -r --model=User
Controller created successfully.

C:\Users\User\Laravel\my_project_01>
```



```
C:\Users\Bashir-BN\Laravel\my_project_01>C:\xampp\php\php.exe artisan route:list --name=post
+-----+-----+-----+-----+-----+-----+
| Domain | Method | URI      | Name    | Action   | Middleware |
+-----+-----+-----+-----+-----+-----+
|        | GET|HEAD| post     | post.index | App\Http\Controllers\PostController@index | web
|        | POST    | post     | post.store | App\Http\Controllers\PostController@store | web
|        | GET|HEAD| post/create | post.create | App\Http\Controllers\PostController@create | web
|        | GET|HEAD| post/{post} | post.show  | App\Http\Controllers\PostController@show  | web
|        | PUT|PATCH| post/{post} | post.update | App\Http\Controllers\PostController@update | web
|        | DELETE  | post/{post} | post.destroy | App\Http\Controllers\PostController@destroy | web
|        | GET|HEAD| post/{post}/edit | post.edit  | App\Http\Controllers\PostController@edit  | web
+-----+-----+-----+-----+-----+-----+
C:\Users\Bashir-BN\Laravel\my_project_01>C:\xampp\php\php.exe artisan route:list --name=user
+-----+-----+-----+-----+-----+-----+
| Domain | Method | URI      | Name    | Action   | Middleware |
+-----+-----+-----+-----+-----+-----+
|        | GET|HEAD| user    | User.index | App\Http\Controllers\UserController@index | web
|        | POST    | user    | user.store | App\Http\Controllers\UserController@store | web
|        | GET|HEAD| user/create | user.create | App\Http\Controllers\UserController@create | web
|        | GET|HEAD| user/{user} | user.show  | App\Http\Controllers\UserController@show  | web
|        | PUT|PATCH| user/{user} | user.update | App\Http\Controllers\UserController@update | web
|        | DELETE  | user/{user} | user.destroy | App\Http\Controllers\UserController@destroy | web
|        | GET|HEAD| user/{user}/edit | user.edit  | App\Http\Controllers\UserController@edit  | web
+-----+-----+-----+-----+-----+-----+
```

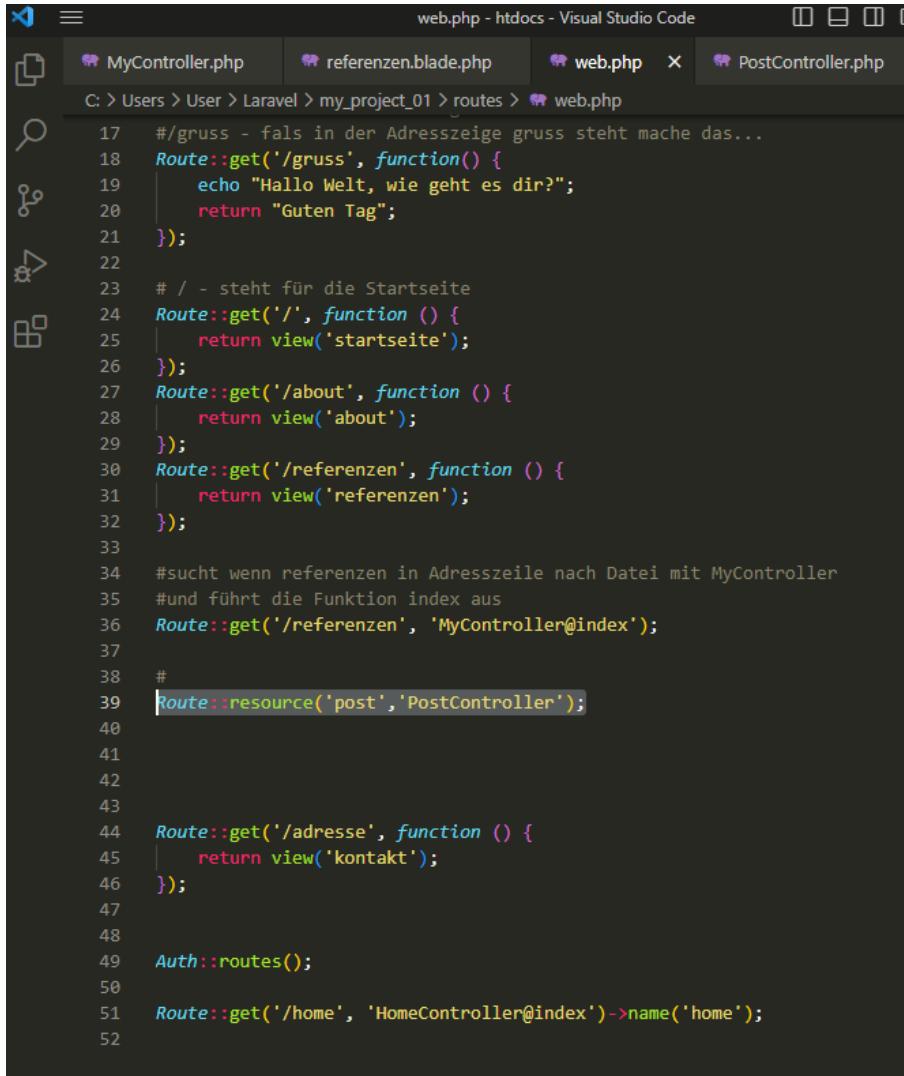


```
UserController.php - htdocs - Visual Studio Code
Controller UserController.php
C:\> Users > User > Laravel > my_project_01 > app > Http > Controllers > UserController.php
UserController.php

1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\User;
6 use Illuminate\Http\Request;
7
8 class UserController extends Controller
9 {
10
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15     */
16     public function index()
17     {
18         //
19     }
20
21     /**
22      * Show the form for creating a new resource.
23      *
24      * @return \Illuminate\Http\Response
25     */
26     public function create()
27     {
28         //
29     }
30
31     /**
32      * Store a newly created resource in storage.
33      *
34      * @param \Illuminate\Http\Request $request
35      * @return \Illuminate\Http\Response
36     */
37     public function store(Request $request)
38     {
39         //
40     }
41
42     /**
43      * Display the specified resource.
44      *
45      * @param \App\User $user
46      * @return \Illuminate\Http\Response
47     */
48     public function show(User $user)
49     {
```

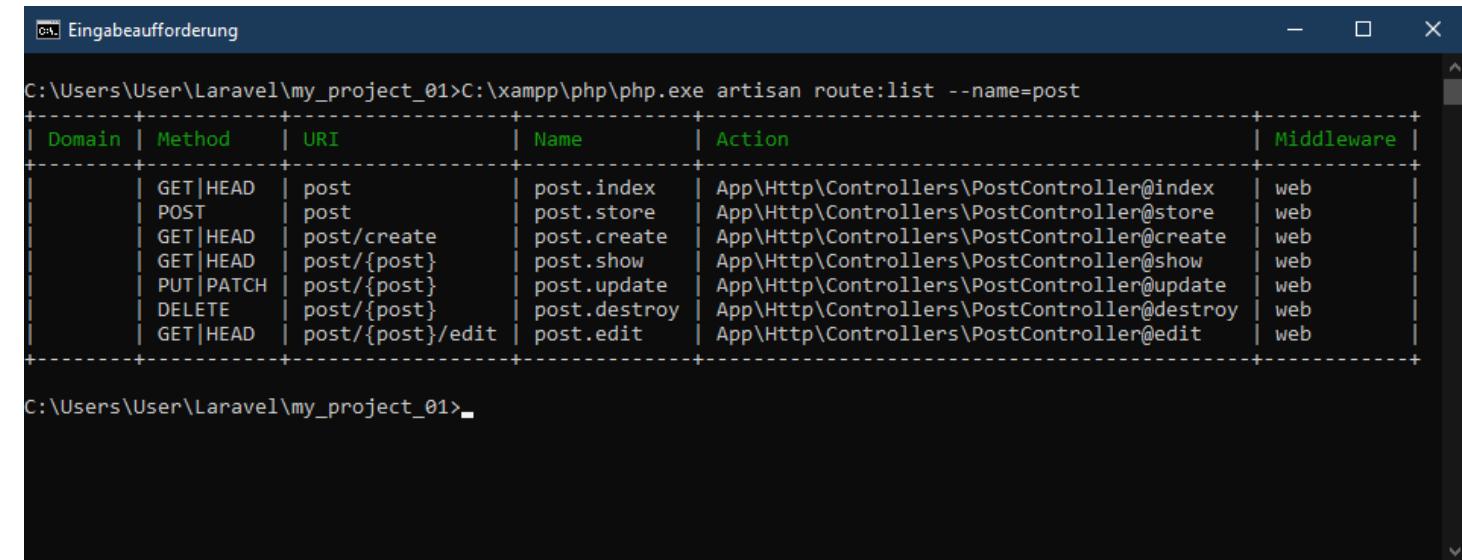
Die Route einer Liste sichtbar machen

C:\xampp\php\php.exe artisan route:list --name=Tabellenname



The screenshot shows the routes/web.php file in Visual Studio Code. The code defines several routes:

```
17 #/gruss - fals in der Adresszeige gruss steht mache das...
18 Route::get('/gruss', function() {
19     echo "Hallo Welt, wie geht es dir?";
20     return "Guten Tag";
21 });
22
23 # / - steht für die Startseite
24 Route::get('/', function () {
25     return view('startseite');
26 });
27 Route::get('/about', function () {
28     return view('about');
29 });
30 Route::get('/referenzen', function () {
31     return view('referenzen');
32 });
33
34 #sucht wenn referenzen in Adresszeile nach Datei mit MyController
35 #und führt die Funktion index aus
36 Route::get('/referenzen', 'MyController@index');
37
38 #
39 Route::resource('post', 'PostController');
40
41
42
43
44 Route::get('/adresse', function () {
45     return view('kontakt');
46 });
47
48
49 Auth::routes();
50
51 Route::get('/home', 'HomeController@index')->name('home');
```



The screenshot shows a terminal window titled "Eingabeaufforderung" displaying the output of the artisan route:list command. The output lists all routes defined in the routes/web.php file, including their names and actions:

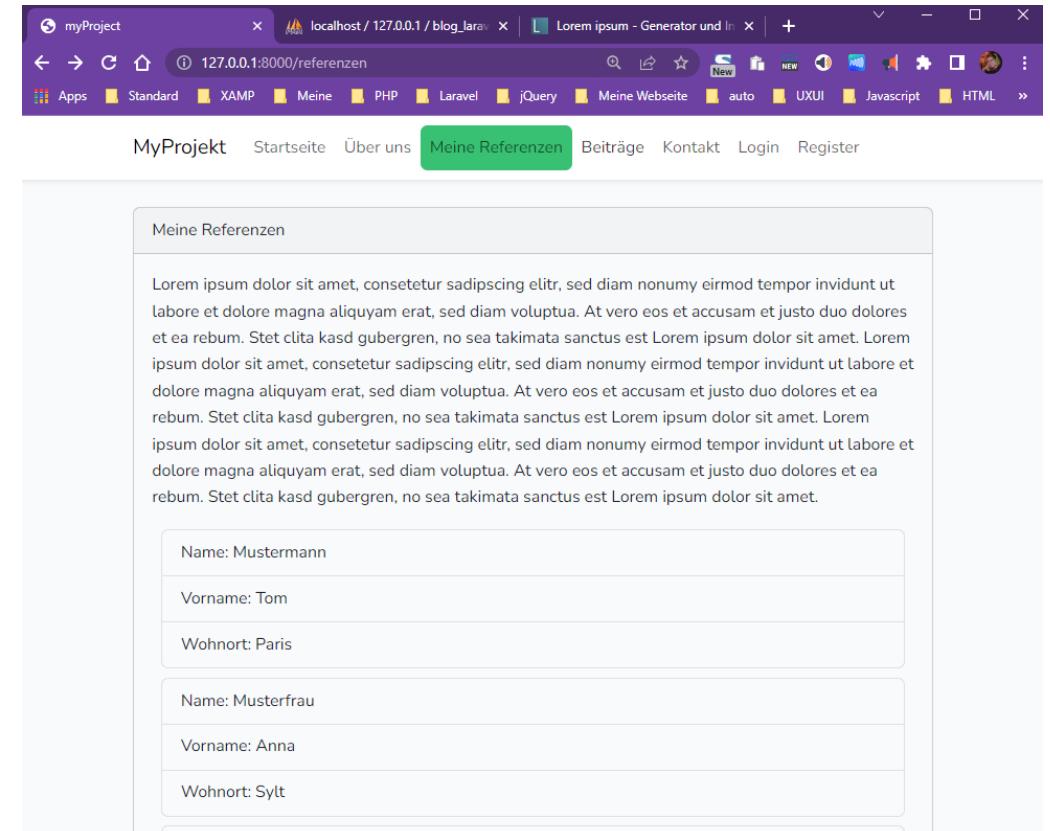
Domain	Method	URI	Name	Action	Middleware
	GET HEAD	post	post.index	App\Http\Controllers\PostController@index	web
	POST	post	post.store	App\Http\Controllers\PostController@store	web
	GET HEAD	post/create	post.create	App\Http\Controllers\PostController@create	web
	GET HEAD	post/{post}	post.show	App\Http\Controllers\PostController@show	web
	PUT PATCH	post/{post}	post.update	App\Http\Controllers\PostController@update	web
	DELETE	post/{post}	post.destroy	App\Http\Controllers\PostController@destroy	web
	GET HEAD	post/{post}/edit	post.edit	App\Http\Controllers\PostController@edit	web

Navigation weiter
anpassen

1) Neuen Navipunkt erstellen

```
<link href="{{ asset('css/app.css') }}" rel="stylesheet">
</head>
<body>
    <div id="app">
        <nav class="navbar navbar-expand-md navbar-light bg-white shadow-sm">
            <div class="container">
                <a class="navbar-brand" href="{{ url('/') }}>MyProjekt</a>
                <!-- {{ config('app.name', 'Laravel') }} -->
                <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="({__('Toggle navigation'))}">
                    <span class="navbar-toggler-icon"></span>
                </button>

                <div class="collapse navbar-collapse" id="navbarSupportedContent">
                    <!-- Left Side Of Navbar -->
                    <ul class="navbar-nav mr-auto">
                        <!-- / steht für Startseite -->
                        <!--request spricht Inhalt an-->
                        <li> <a class="nav-link" {{ Request::is('/') ? 'btn bg-success' : ''}}>Startseite</a></li>
                        <li> <a class="nav-link" {{ Request::is('about') ? 'btn bg-success' : ''}}>{{__('Über uns')}}</a></li>
                        <li> <a class="nav-link" {{ Request::is('referenzen') ? 'btn bg-success' : ''}}>{{__('Meine Referenzen')}}</a></li>
                    <!-- Right Side Of Navbar -->
                    <ul class="navbar-nav ml-auto">
                        <!-- Authentication Links -->
                        @guest
                            <li class="nav-item">
                                <a class="nav-link" href="{{ route('login') }}>{{ __('Login') }}</a>
                            </li>
                            @if (Route::has('register'))
                                <li class="nav-item">
                                    <a class="nav-link" href="{{ route('register') }}>{{ __('Register') }}</a>
                                </li>
                            @endif
                        @endguest
                </div>
            </div>
        </nav>
    </div>
</body>
```

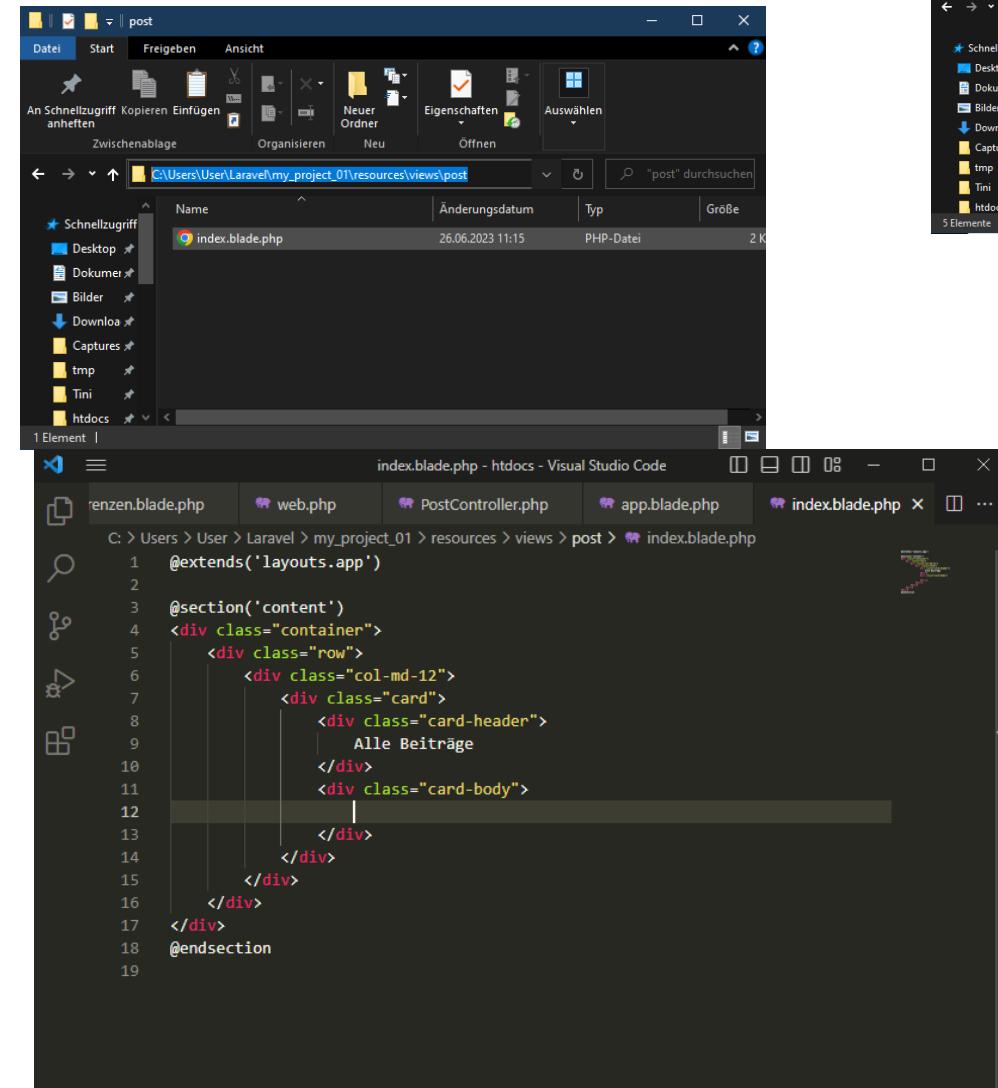


2) Test ob Seite geht, ohne das sie angelegt ist (noch keine yyy.blade.php ist erstellt)

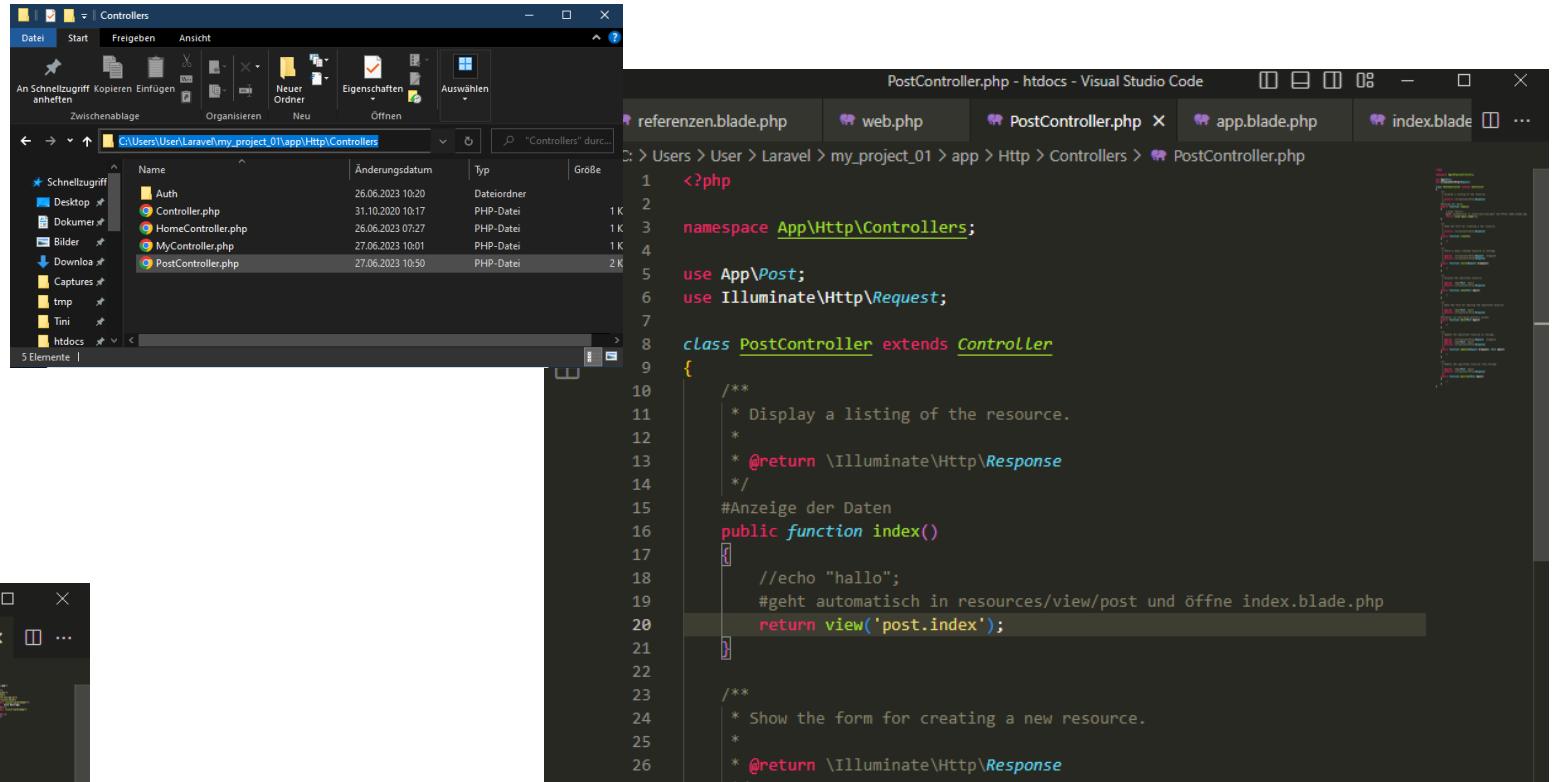
The image shows a screenshot of a development environment. On the left, the Visual Studio Code interface is visible, displaying the file `PostController.php`. The code defines a controller class `PostController` that contains three methods: `index()`, `create()`, and `store()`. The `index()` method simply outputs the string "hallo". The browser window on the right shows the URL `127.0.0.1:8000/post`, which displays the word "hallo" in the page content.

```
<?php
namespace App\Http\Controllers;
use App\Post;
use Illuminate\Http\Request;
class PostController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    #Anzeige der Daten
    public function index()
    {
        echo "hallo";
    }
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        //
    }
    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request)
    {
```

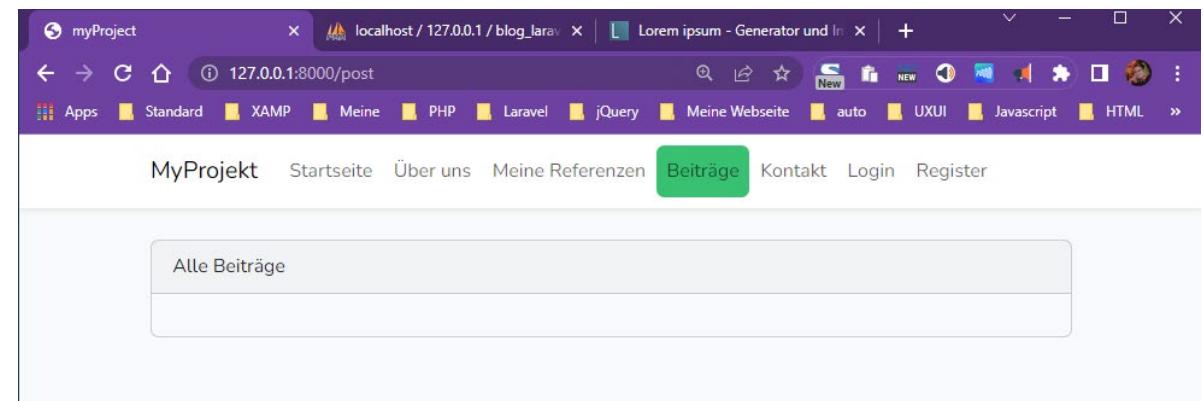
3) Seite zum neuen Navipunkt anlegen



```
index.blade.php - htdocs - Visual Studio Code
C: > Users > User > Laravel > my_project_01 > resources > views > post > index.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-12">
7       <div class="card">
8         <div class="card-header">
9           Alle Beiträge
10        </div>
11        <div class="card-body">
12          <table border="1">
13            <thead>
14              <tr>
15                <th>Titel</th>
16                <th>Autorenname</th>
17                <th>Datum</th>
18              </tr>
19            </thead>
20            <tbody>
21              <tr>
22                <td>Post 1</td>
23                <td>John Doe</td>
24                <td>2023-06-27</td>
25              </tr>
26              <tr>
27                <td>Post 2</td>
28                <td>Jane Doe</td>
29                <td>2023-06-28</td>
30              </tr>
31            </tbody>
32          </table>
33        </div>
34      </div>
35    </div>
36  </div>
37 </div>
38 @endsection
```



```
PostController.php - htdocs - Visual Studio Code
referenzen.blade.php web.php PostController.php X app.blade.php index.blade.php ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Post;
6 use Illuminate\Http\Request;
7
8 class PostController extends Controller
9 {
10
11   /**
12    * Display a listing of the resource.
13    *
14    * @return \Illuminate\Http\Response
15   */
16   #Anzeige der Daten
17   public function index()
18   {
19     //echo "hallo";
20     #geht automatisch in resources/view/post und öffne index.blade.php
21     return view('post.index');
22
23   /**
24    * Show the form for creating a new resource.
25    *
26    * @return \Illuminate\Http\Response
27   }
28 }
```



Dynamische Seiten erstellen

uebung01 > routes > [web.php](#)

```

1 web.php
2
3 Route::resource('wert','WertController');
4 Route::get('wert/{wert}', function(int $nummer){
5     $erg='';
6     if(is_int($nummer)){
7         for ($i = 1; $i <= $nummer; $i++){
8             $erg .= "<p>$i x $i = ". $i*$i."</p>";
9         }
10    return $erg;
11 });
12
13 Route::resource('zahl','ZahlController');
14 Route::get('zahl/{zahl}', function(int $nummer){
15     $erg='';
16     if(is_int($nummer)){
17         $erg .= "<p>". $nummer." x ". $nummer." = ". $nummer * $nummer."</p>";
18     }
19     return $erg;
20 });
21
22 Route::resource('user','UserController');
23 Route::get('user/{age}/{user}', function($age,$name){
24     $erg='';
25
26     $erg .= "Hallo ". $name."<br />";
27     $erg .= "Alter: ".$age."j";
28
29     return $erg;
30 });
31
32 Auth::routes();
33
34 Route::get('/home', 'HomeController@index')->name('home');
35
36
37
38
39
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63

```

127.0.0.1:8000/wert/2

1 x 1 = 1*1
2 x 2 = 2*2
1 x 1 = 1*1
2 x 2 = 2*2

127.0.0.1:8000/wert/5

1 x 1 = 1*1
2 x 2 = 2*2
3 x 3 = 3*3
4 x 4 = 4*4
5 x 5 = 5*5

index.blade.php

```

1 uebung01 > resources > views > wert > index.blade.php
2 @extends('layouts.app')
3
4 @section('title','Wert')
5
6 @section('content')
7 <div class="container text-center">
8     <div class="row">
9         <div class="col-md-12 fs-2" style="background:var(--bs-gray-dark);">
10            <div class="text-start m-4" >
11                <a href="/wert/2" class="btn btn-success mt-2">2</a>
12                <a href="/wert/5" class="btn btn-success mt-2">5</a>
13            </div>
14        </div>
15    </div>
16 </div>
17 @endsection
18
19

```

127.0.0.1:8000/zahl/2

2 x 2 = 4

127.0.0.1:8000/zahl/6

6 x 6 = 36

index.blade.php

```

1 uebung01 > resources > views > zahl > index.blade.php
2 @extends('layouts.app')
3
4 @section('title','Zahl')
5
6 @section('content')
7 <div class="container text-center">
8     <div class="row">
9         <div class="col-md-12 fs-2" style="background:var(--bs-gray-dark);">
10            <div class="text-start m-4" >
11                <a href="/zahl/2" class="btn btn-success mt-2">2</a>
12                <a href="/zahl/6" class="btn btn-success mt-2">6</a>
13            </div>
14        </div>
15    </div>
16 </div>
17 @endsection
18
19

```

index.blade.php

```

1 uebung01 > resources > views > user > index.blade.php
2 @extends('layouts.app')
3
4 @section('title','Wert')
5
6 @section('content')
7 <div class="container text-center">
8     <div class="row">
9         <div class="col-md-12 fs-2" style="background:var(--bs-gray-dark);">
10            <div class="text-start m-4" >
11                <a href="user/23/hugo" class="btn btn-success mt-2">hugo</a>
12                <a href="user/35/anna" class="btn btn-success mt-2">anna</a>
13            </div>
14        </div>
15    </div>
16 </div>
17 @endsection
18
19

```

myProject

- Home
- About
- Portfolio
- Kontakt
- Wert
- Zahl
- User

hugo anna

127.0.0.1:8000/user/23/hugo

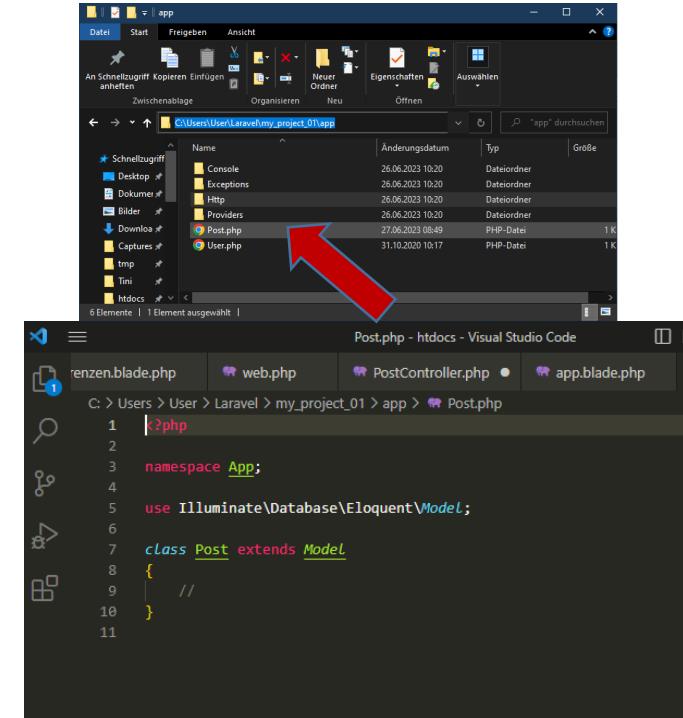
127.0.0.1:8000/user/35/anna

Haloo hugo
Alter: 23j

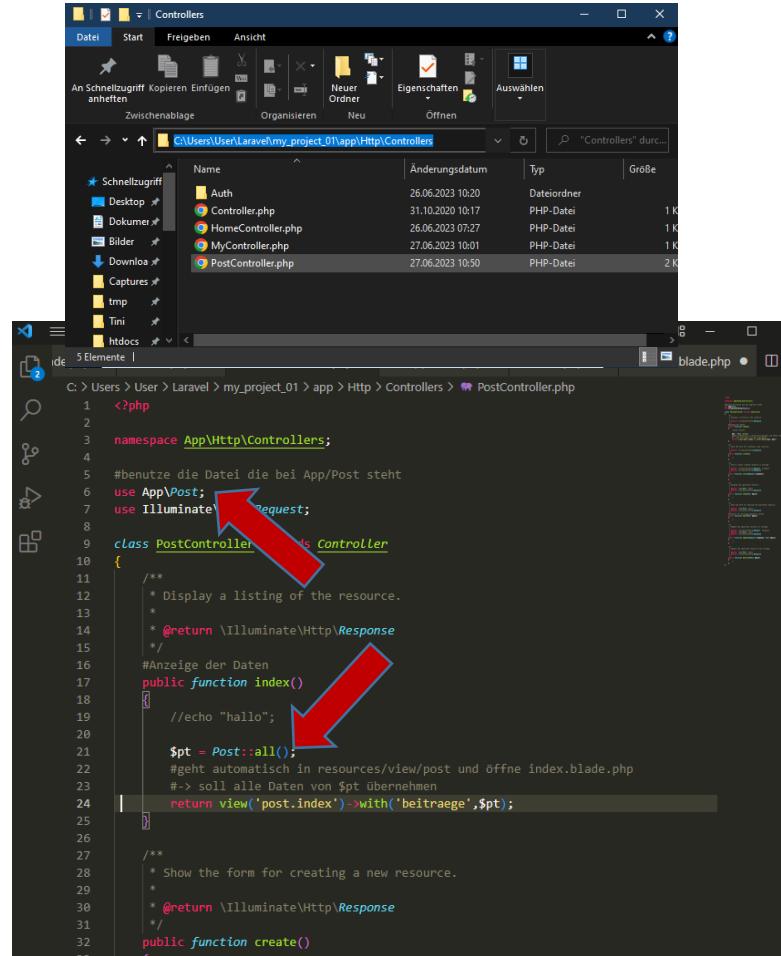
Haloo anna
Alter: 35j

Daten aus
einer DB anzeigen

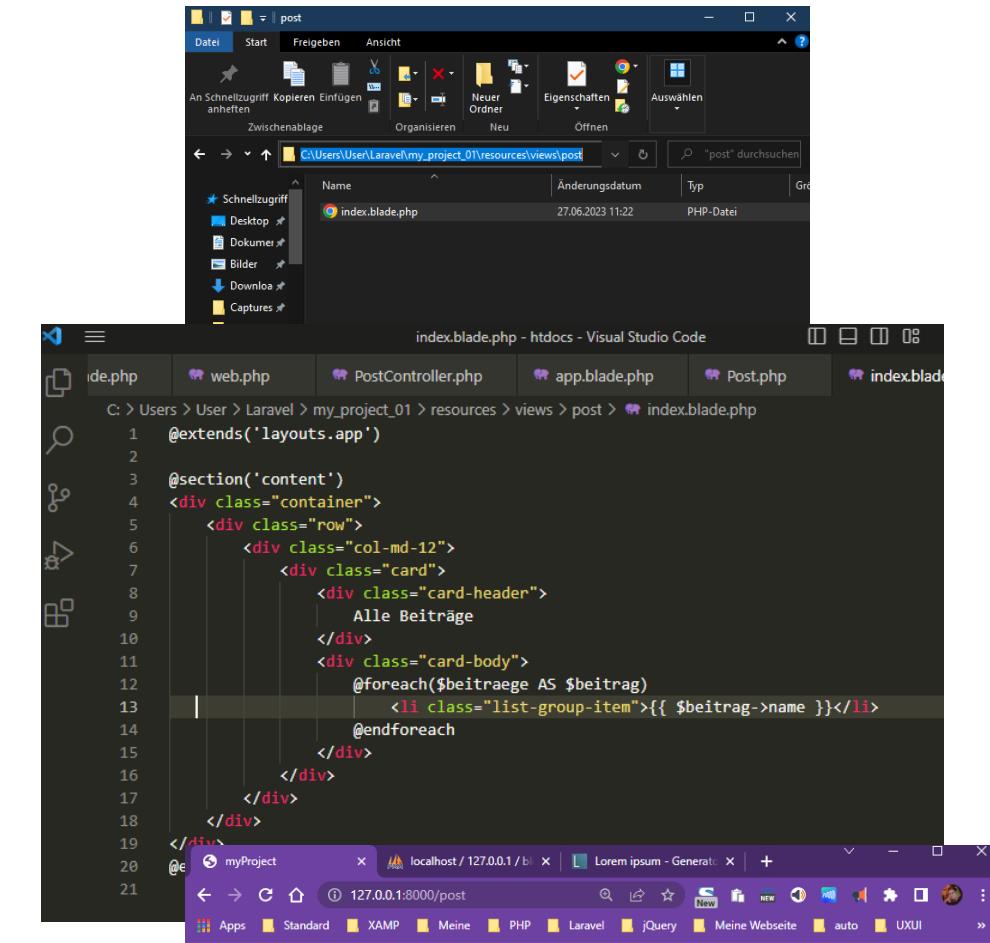
1) Daten aus DB anzeigen in neuer Seite



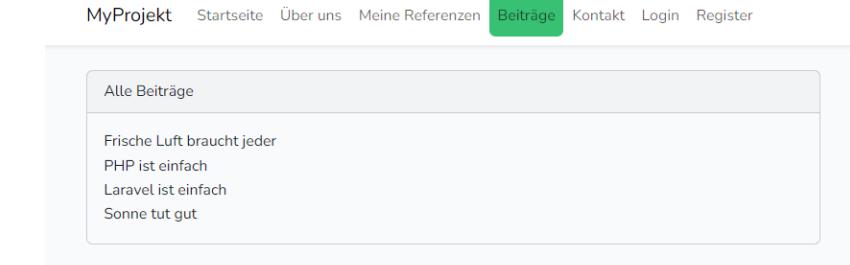
```
Post.php - htdocs - Visual Studio Code
C: > Users > User > Laravel > my_project_01 > app > Post.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Post extends Model
8 {
9     //
10 }
11
```



```
PostController.php - htdocs - Visual Studio Code
C: > Users > User > Laravel > my_project_01 > app > Http > Controllers > PostController.php
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use App\Post;
6 use Illuminate\Http\Request;
7
8 class PostController extends Controller
9 {
10
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16
17     #Anzeige der Daten
18     public function index()
19     {
20         //echo "hallo";
21
22         $pt = Post::all();
23         #geht automatisch in resources/view/post und öffne index.blade.php
24         #=> soll alle Daten von $pt übernehmen
25         return view('post.index')->with('beitraege',$pt);
26
27     }
28
29     /**
30      * Show the form for creating a new resource.
31      *
32      * @return \Illuminate\Http\Response
33      */
34
35     public function create()
36     {
37
38     }
39
40 }
```

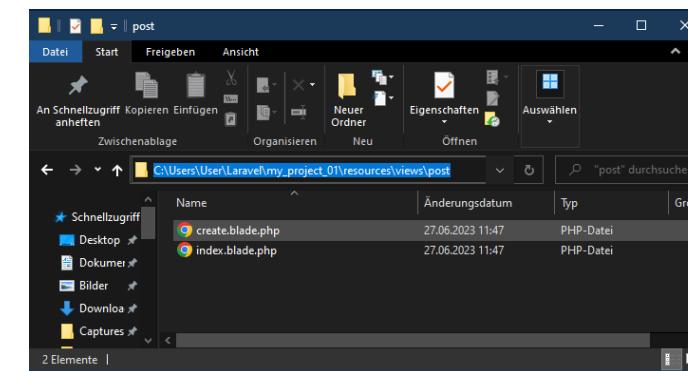


```
index.blade.php - htdocs - Visual Studio Code
ide.php web.php PostController.php app.blade.php Post.php index.blade.php
C: > Users > User > Laravel > my_project_01 > resources > views > post > index.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     <div class="row">
6         <div class="col-md-12">
7             <div class="card">
8                 <div class="card-header">
9                     Alle Beiträge
10                </div>
11                <div class="card-body">
12                    @foreach($beitraege AS $beitrag)
13                        <li class="list-group-item">{{ $beitrag->name }}</li>
14                    @endforeach
15                </div>
16            </div>
17        </div>
18    </div>
19 </div>
20
21 @endsection
```



Daten in
DB eintragen

Neuer Beitrag/Reiter/neuen Datensatz anlegen



A screenshot of Visual Studio Code showing the 'index.blade.php' file. The code is a Blade template:

```
@extends('layouts.app')
@section('content')
    <div class="container">
        <div class="row">
            <div class="col-md-12">
                <div class="card">
                    <div class="card-header">
                        Alle Beiträge
                    </div>
                    <div class="card-body">
                        <ul class="list-group">
                            @foreach($beitraege AS $beitrag)
                                <li class="list-group-item">{{ $beitrag->name }}</li>
                            @endforeach
                        </ul>
                        <a href="post/create" class="btn btn-success mt-2">Neuer Beitrag anlegen</a>
                    </div>
                </div>
            </div>
        </div>
    </div>
@endsection
```

A screenshot of a browser window titled 'MyProjekt'. The URL is '127.0.0.1:8000/post'. The page displays a list of posts with a 'Neuer Beitrag anlegen' button at the bottom.

Post
Frische Luft braucht jeder
PHP ist einfach
Laravel ist einfach
Sonne tut gut

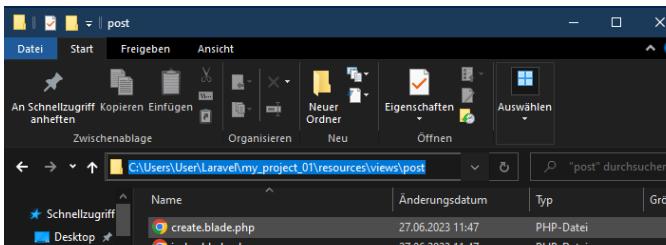
At the bottom of the page is a green button labeled 'Neuer Beitrag anlegen'.

A screenshot of Visual Studio Code showing the 'PostController.php' file. The code is a Laravel controller:

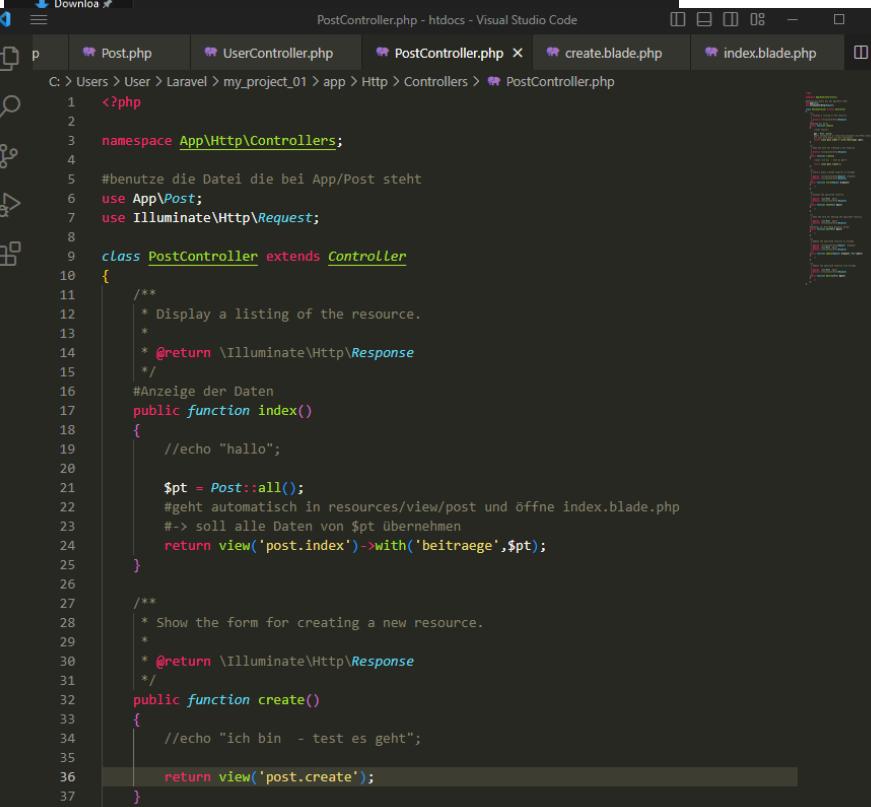
```
<?php
namespace App\Http\Controllers;
use App\Post;
use Illuminate\Http\Request;
class PostController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    #Anzeige der Daten
    public function index()
    {
        //echo "Hallo";
        $pt = Post::all();
        #geht automatisch in resources/view/post und öffne index.blade.php
        #-> soll alle Daten von $pt übernehmen
        return view('post.index')->with('beitraege',$pt);
    }
    /**
     * Show the form for creating a new resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function create()
    {
        echo "ich bin - test es geht";
    }
    /**
     * Store a newly created resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @return \Illuminate\Http\Response
     */
}
```

A screenshot of a browser window showing the result of the 'create' action. The URL is '127.0.0.1:8000/post/create'. The page displays the text 'ich bin - test es geht'.

Neuer Beitrag/Reiter/neuen Datensatz anlegen

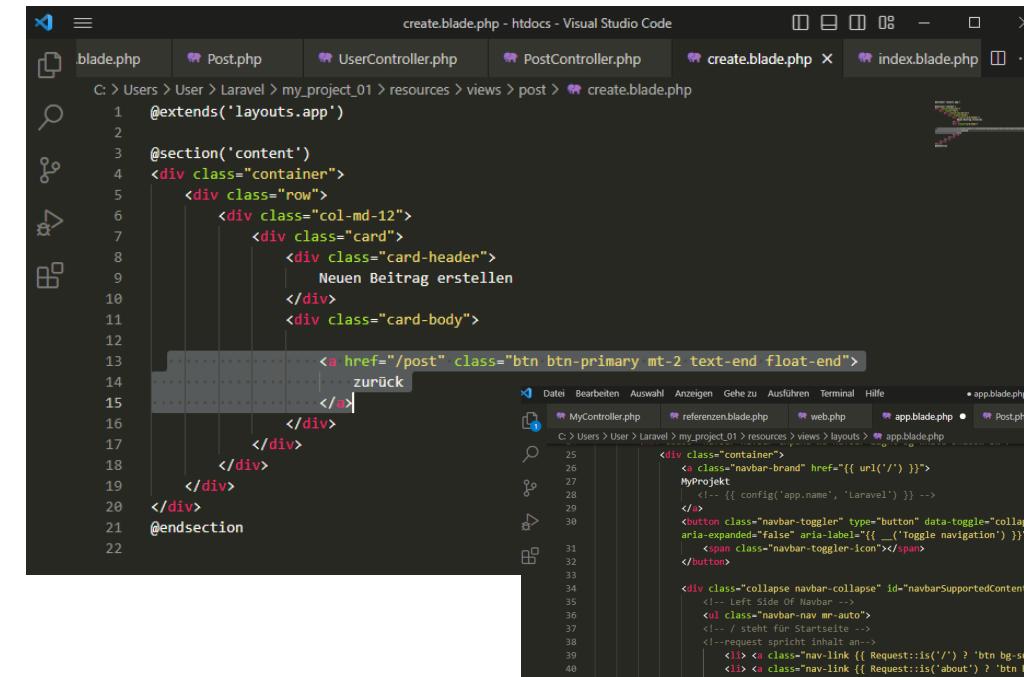


Screenshot of a Windows File Explorer window showing the contents of the 'post' directory. It contains two files: 'create.blade.php' and 'index.blade.php'. Both files were created on 27.06.2023 at 11:47 and are PHP files.



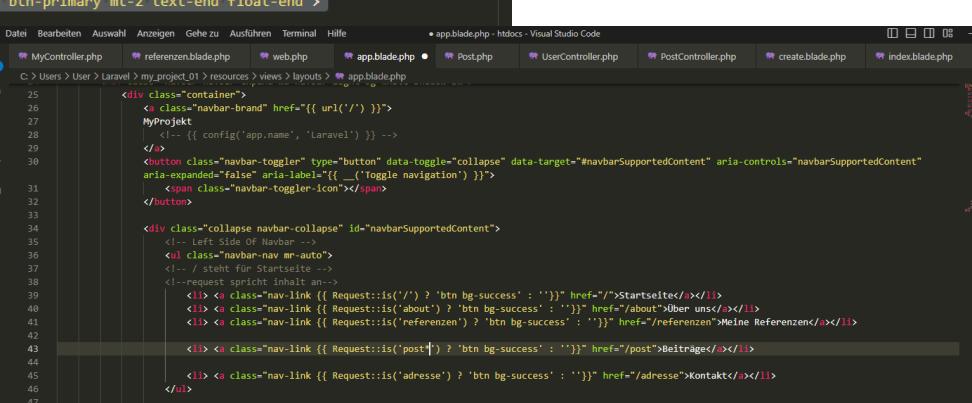
Screenshot of Visual Studio Code showing the 'PostController.php' file. The code defines a 'PostController' class extending 'Controller'. It has two methods: 'index()' which returns a view named 'post.index' and 'create()' which returns a view named 'post.create'. The 'index()' method also calls 'Post::all()' to retrieve all posts.

```
1 <?php
2
3 namespace App\Http\Controllers;
4
5 #benutze die Datei die bei App/Post steht
6 use App\Post;
7 use Illuminate\Http\Request;
8
9 class PostController extends Controller
10 {
11     /**
12      * Display a listing of the resource.
13      *
14      * @return \Illuminate\Http\Response
15      */
16     #Anzeige der Daten
17     public function index()
18     {
19         //echo "hallo";
20
21         $pt = Post::all();
22         #geht automatisch in resources/view/post und öffne index.blade.php
23         #-> soll alle Daten von $pt übernehmen
24         return view('post.index')->with('beitraege',$pt);
25     }
26
27     /**
28      * Show the form for creating a new resource.
29      *
30      * @return \Illuminate\Http\Response
31      */
32     public function create()
33     {
34         //echo "ich bin - test es geht";
35
36         return view('post.create');
37     }
38 }
```



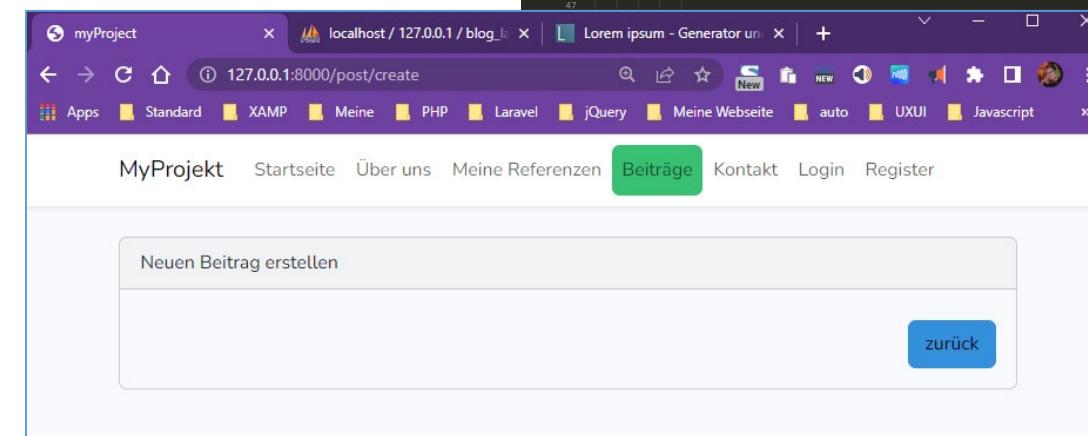
Screenshot of Visual Studio Code showing the 'create.blade.php' file. The code is a Blade template. It starts with '@extends('layouts.app')' and '@section('content')'. Inside the section, it displays a heading 'Neuen Beitrag erstellen' and a button 'zurück'. The template ends with '@endsection'.

```
1 @extends('layouts.app')
2
3 @section('content')
4     <div class="container">
5         <div class="row">
6             <div class="col-md-12">
7                 <div class="card">
8                     <div class="card-header">
9                         Neuen Beitrag erstellen
10                    </div>
11                     <div class="card-body">
12                         <a href="/post" class="btn btn-primary mt-2 text-end float-end">
13                             zurück
14                         </a>
15                     </div>
16                 </div>
17             </div>
18         </div>
19     </div>
20     @endsection
```



Screenshot of Visual Studio Code showing the 'navbar.blade.php' file. This is part of a larger navigation bar. It includes a logo 'MyProject', a toggle button for the sidebar, and a dropdown menu with items like 'Startseite', 'Über uns', 'Meine Referenzen', and 'Beiträge'. The 'Beiträge' item is highlighted in green.

```
1 <div class="container">
2     <!-- navbar-brand -->
3     <a href="{{ url('/') }}">MyProject</a>
4     <!-- {{ config('app.name', 'Laravel') }} -->
5     <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="{{ __('Toggle navigation') }}>
6         <span class="navbar-toggler-icon"></span>
7     </button>
8
9     <div class="collapse navbar-collapse" id="navbarSupportedContent">
10        <!-- Left Side Of Navbar -->
11        <ul class="navbar-nav mr-auto">
12            <!-- / steht für Startseite -->
13            <!-- request spricht Inhalt an -->
14            <li><a class="nav-link" {{ Request::is('/') ? 'btn bg-success' : ''}} href="/startseite">Startseite</a></li>
15            <li><a class="nav-link" {{ Request::is('about') ? 'btn bg-success' : ''}} href="/about">Über uns</a></li>
16            <li><a class="nav-link" {{ Request::is('referenzen') ? 'btn bg-success' : ''}} href="/referenzen">Meine Referenzen</a></li>
17            <li><a class="nav-link" {{ Request::is('post') ? 'btn bg-success' : ''}} href="/post">Beiträge</a></li>
18            <li><a class="nav-link" {{ Request::is('adresse') ? 'btn bg-success' : ''}} href="/adresse">Kontakt</a></li>
19        </ul>
20    </div>
```



Datenbank posts

Server: 127.0.0.1 » Datenbank: blog_laravel_webfull02 » Tabelle: posts

Anzeigen Struktur SQL Suche Einfügen Exportieren Importieren Rechte Operationen Mehr

Tabellenstruktur Beziehungsansicht

#	Name	Typ	Kollation	Attribute	Null	Standard	Kommentare	Extra	Aktion
1	id	int(10)		UNSIGNED	Nein	kein(e)		AUTO_INCREMENT	Bearbeiten Löschen Mehr
2	name	varchar(255)	utf8mb4_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
3	beschreibung	text	utf8mb4_unicode_ci		Nein	kein(e)			Bearbeiten Löschen Mehr
4	created_at	timestamp			Ja	NULL			Bearbeiten Löschen Mehr
5	updated_at	timestamp			Ja	NULL			Bearbeiten Löschen Mehr

Alle auswählen markierte: Anzeigen Bearbeiten Löschen Primärschlüssel Unique Index Räumlich

Volltext Zu zentralen Spalten hinzufügen Zentrale Spalten entfernen

Zeige Datensätze 0 - 7 (8 insgesamt. Die Abfrage dauerte 0.0006 Sekunden.) [beschreibung: [BLOB - 9 B]... [BLOB - 11 B]...]

SELECT * FROM `posts` ORDER BY `beschreibung` ASC

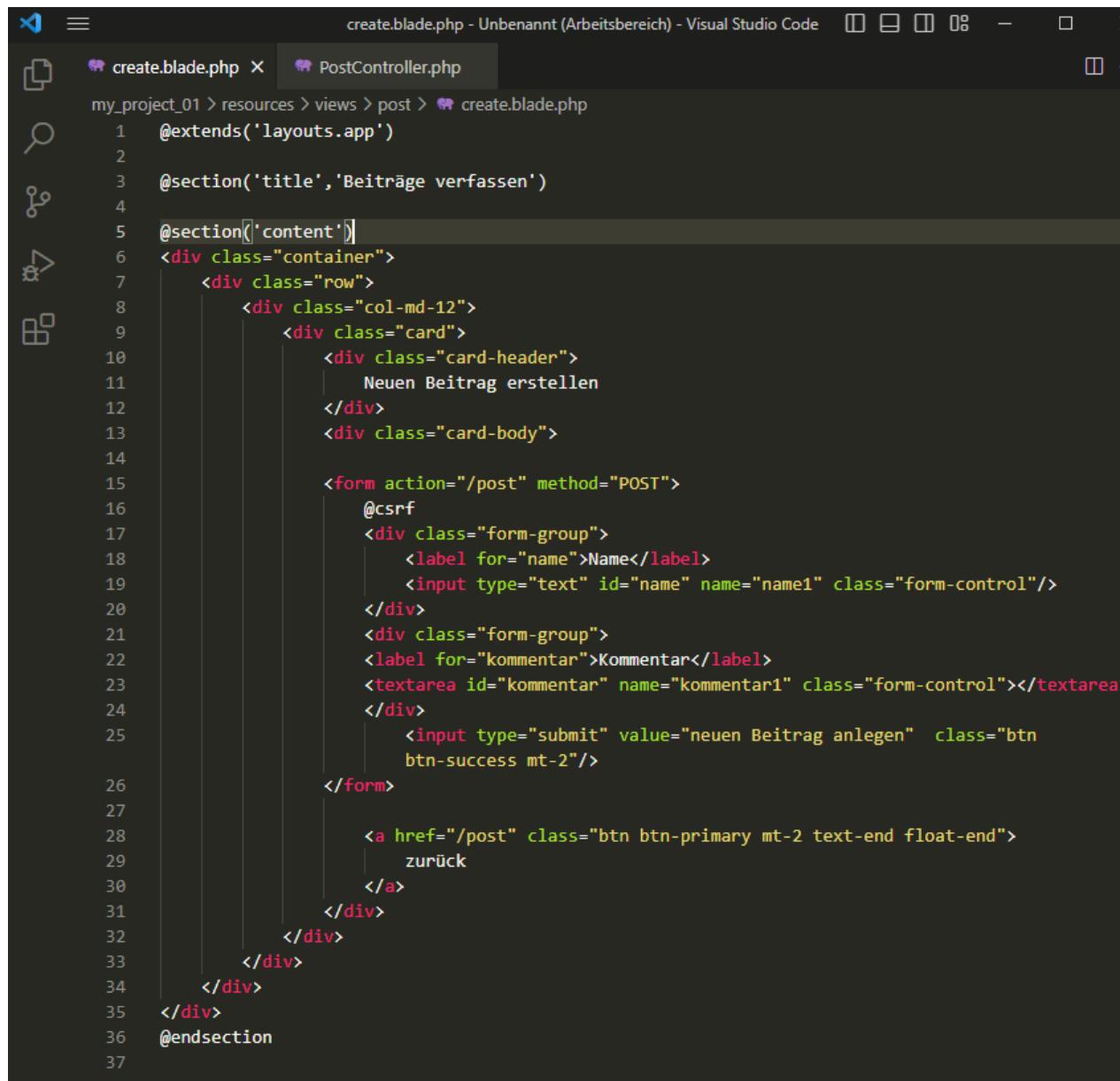
Messen [Inline bearbeiten] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Alles anzeigen Anzahl der Datensätze: 25 Zeilen filtern Diese Tabelle durchsuchen Nach Schlüssel sortieren: keine

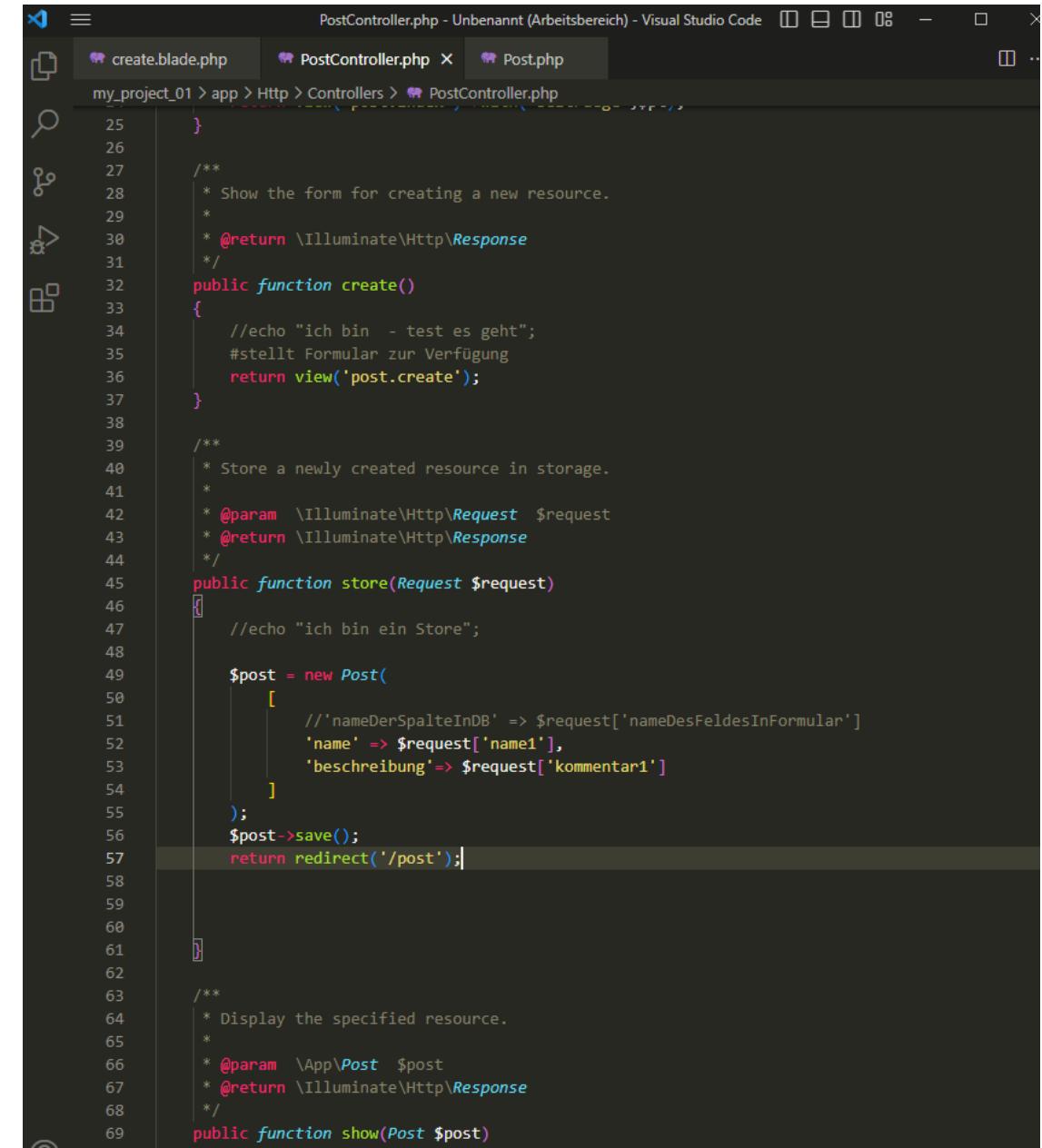
		id	name	beschreibung	created_at	updated_at
Bearbeiten	Kopieren	8	Klaus	Halloooo	2023-06-28 12:15:09	2023-06-28 12:15:09
Bearbeiten	Kopieren	7	Tini	ich mag dich sehr das ist mein Ernst	2023-06-28 09:19:24	2023-06-28 09:19:24
Bearbeiten	Kopieren	5	Der Lenz ist da	Komme gleich raus	2023-06-28 08:17:53	2023-06-28 08:17:53
Bearbeiten	Kopieren	1	Frische Luft braucht jeder	Lorem ipsum dolor sit amet, consectetur adipisicing...	NULL	NULL
Bearbeiten	Kopieren	2	PHP ist einfach	Lorem ipsum dolor sit amet, consectetur adipisicing...	NULL	NULL
Bearbeiten	Kopieren	3	Laravel ist einfach	Sonnenblume	NULL	NULL
Bearbeiten	Kopieren	4	Sonne tut gut	Sonnenblume	NULL	NULL
Bearbeiten	Kopieren	6	Guten Morgen Welt	Welt	2023-06-28 08:20:03	2023-06-28 08:20:03

Alle auswählen markierte: Bearbeiten Kopieren Löschen Exportieren

1) Create Formular



```
create.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code
my_project_01 > resources > views > post > create.blade.php
1 @extends('layouts.app')
2
3 @section('title','Beiträge verfassen')
4
5 @section('content')
6     <div class="container">
7         <div class="row">
8             <div class="col-md-12">
9                 <div class="card">
10                     <div class="card-header">
11                         Neuen Beitrag erstellen
12                     </div>
13                     <div class="card-body">
14
15                         <form action="/post" method="POST">
16                             @csrf
17                             <div class="form-group">
18                                 <label for="name">Name</label>
19                                 <input type="text" id="name" name="name1" class="form-control"/>
20                             </div>
21                             <div class="form-group">
22                                 <label for="kommentar">Kommentar</label>
23                                 <textarea id="kommentar" name="kommentar1" class="form-control"></textarea>
24                             </div>
25                             <input type="submit" value="neuen Beitrag anlegen" class="btn btn-success mt-2"/>
26                         </form>
27
28                         <a href="/post" class="btn btn-primary mt-2 text-end float-end">
29                             zurück
30                         </a>
31                     </div>
32                 </div>
33             </div>
34         </div>
35     @endsection
```



```
PostController.php - Unbenannt (Arbeitsbereich) - Visual Studio Code
my_project_01 > app > Http > Controllers > PostController.php
25 }
26
27 /**
28 * Show the form for creating a new resource.
29 *
30 * @return \Illuminate\Http\Response
31 */
32 public function create()
33 {
34     //echo "ich bin - test es geht";
35     #stellt Formular zur Verfügung
36     return view('post.create');
37 }
38
39 /**
40 * Store a newly created resource in storage.
41 *
42 * @param \Illuminate\Http\Request $request
43 * @return \Illuminate\Http\Response
44 */
45 public function store(Request $request)
46 {
47     //echo "ich bin ein Store";
48
49     $post = new Post(
50         [
51             // 'nameDerSpalteInDB' => $request['nameDesFeldesInFormular']
52             'name' => $request['name1'],
53             'beschreibung' => $request['kommentar1']
54         ]
55     );
56     $post->save();
57     return redirect('/post');
58 }
59
60 }
61
62 /**
63 * Display the specified resource.
64 *
65 * @param \App\Post $post
66 * @return \Illuminate\Http\Response
67 */
68 public function show(Post $post)
69 {
```

2) CSRF - Feld erzeugen

The screenshot shows a development environment with two main windows:

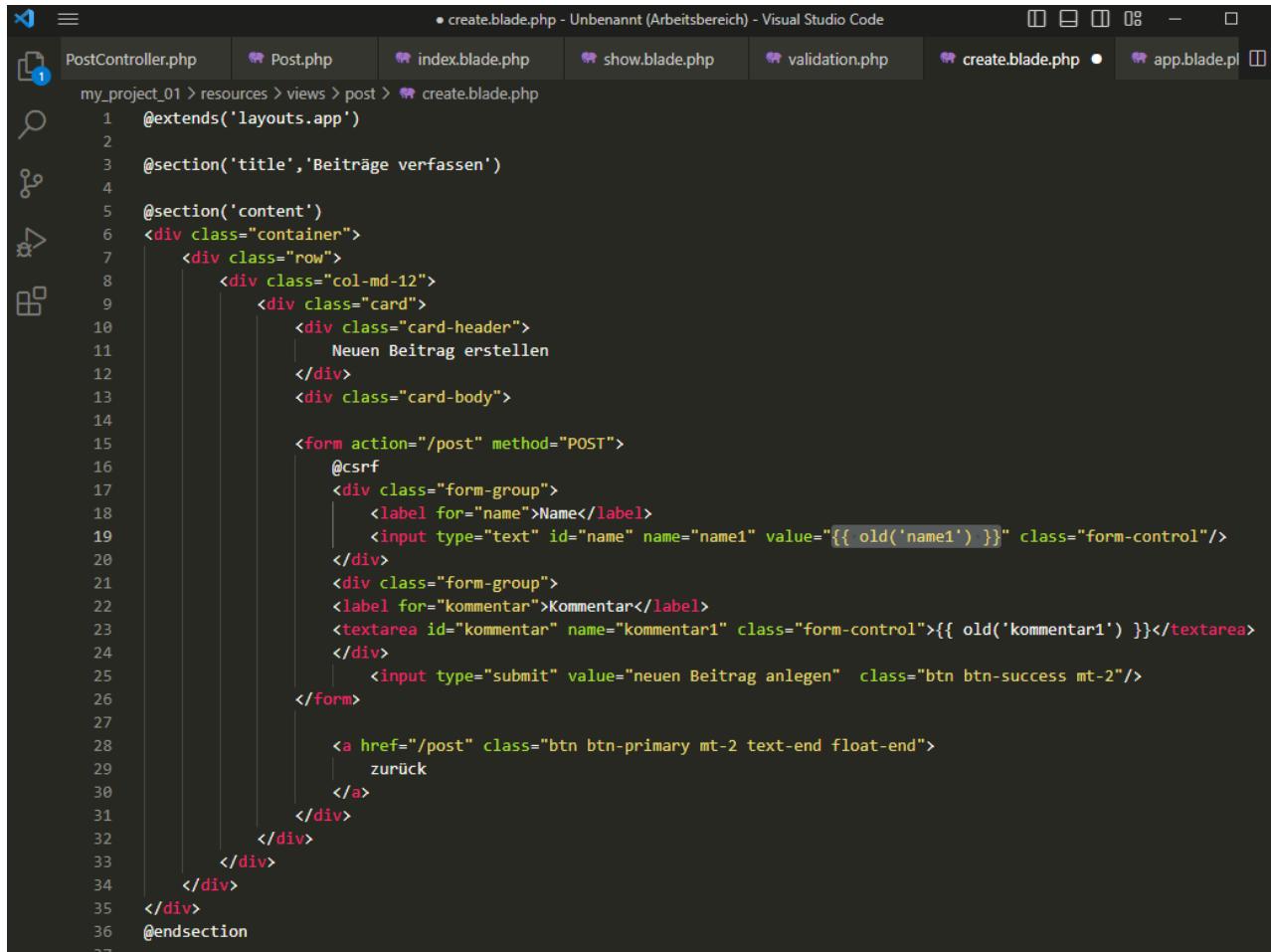
- Left Window (Browser Preview):** Displays a form titled "Neuen Beitrag erstellen". It has fields for "Name" and "Kommentar", and a green button labeled "neuen Beitrag anlegen". A blue button labeled "zurück" is also visible.
- Right Window (Code Editor):** Shows the `create.blade.php` file from the Laravel `resources/views/post` directory. The code defines a form with a CSRF token input field (`<input type="hidden" name="_token" value="DYSzfupKB5YN5Onk7csqeug4caxt60ssbvgku11" =`) and a submit button. The editor interface includes tabs for "create.blade.php" and "PostController.php", and various toolbars for file operations and code navigation.

```
create.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code
```

```
my_project_01 > resources > views > post > create.blade.php
```

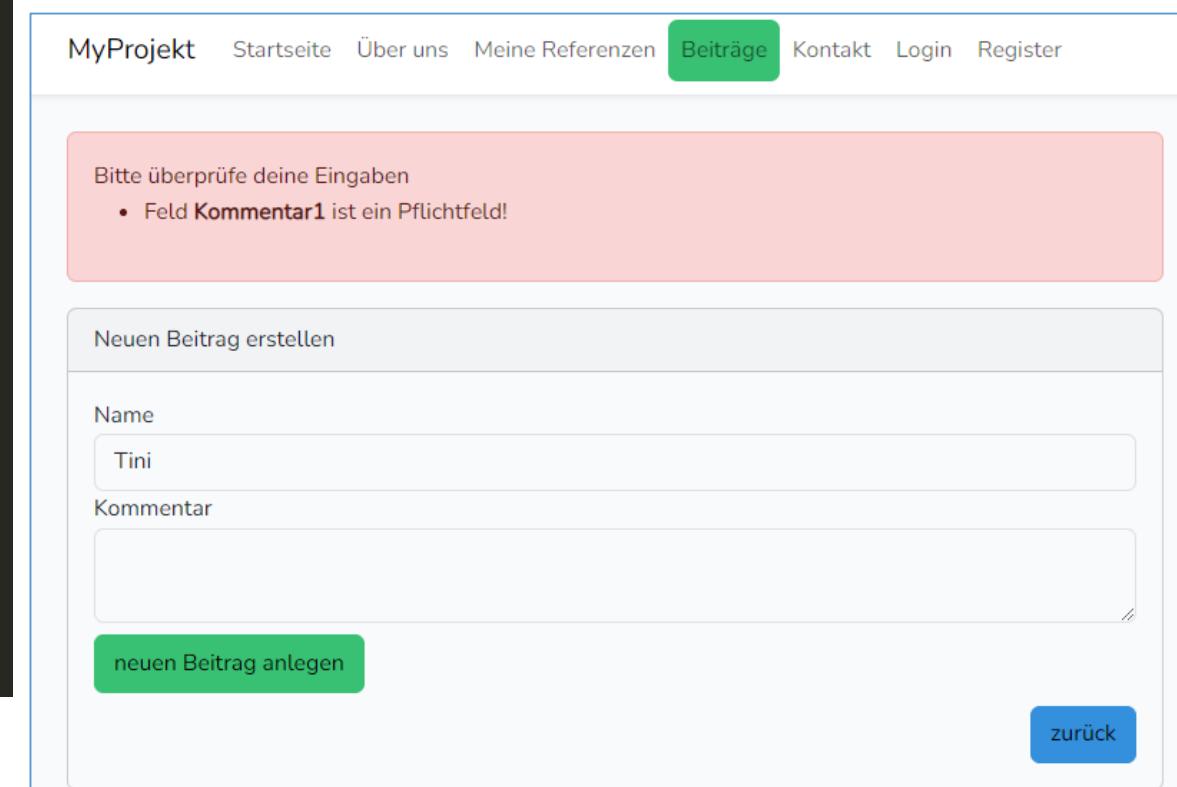
```
1 @extends('layouts.app')
2
3 @section('title', 'Beiträge verfassen')
4
5 @section('content')
6 <div class="container">
7   <div class="row">
8     <div class="col-md-12">
9       <div class="card">
10         <div class="card-header"> Neuen Beitrag erstellen </div>
11         <div class="card-body">
12           <form action="/post" method="POST">
13             <input type="hidden" name="_token" value="DYSzfupKB5YN5Onk7csqeug4caxt60ssbvgku11" =>
14             <div class="form-group"></div>
15             <div class="form-group"></div>
16             <input type="submit" value="neuen Beitrag anlegen" class="btn btn-success mt-2">
17           </form>
18           <a href="/post" class="btn btn-primary mt-2 text-end float-end"> zurück </a>
19         </div>
20       </div>
21     </div>
22   </div>
23 </div>
24
25 <form action="/post" method="POST">
26   @csrf
27   <div class="form-group">
28     <label for="name">Name</label>
29     <input type="text" id="name" name="name1" class="form-control"/>
30   </div>
31   <div class="form-group">
32     <label for="kommentar">Kommentar</label>
33     <textarea id="kommentar" name="kommentar1" class="form-control"></textarea>
34   </div>
35   <input type="submit" value="neuen Beitrag anlegen" class="btn btn-success mt-2"/>
36 </form>
37
38 <a href="/post" class="btn btn-primary mt-2 text-end float-end">
39   zurück
40 </a>
41 </div>
42 </div>
43 </div>
44 </div>
45 @endsection
```

2a) eingegebene Daten behalten



```
PostController.php Post.php index.blade.php show.blade.php validation.php create.blade.php app.blade.php
my_project_01 > resources > views > post > create.blade.php
1 @extends('layouts.app')
2
3 @section('title','Beiträge verfassen')
4
5 @section('content')
6 <div class="container">
7   <div class="row">
8     <div class="col-md-12">
9       <div class="card">
10         <div class="card-header">
11           Neuen Beitrag erstellen
12         </div>
13         <div class="card-body">
14
15           <form action="/post" method="POST">
16             @csrf
17             <div class="form-group">
18               <label for="name">Name</label>
19               <input type="text" id="name" name="name1" value="{{ old('name1') }}" class="form-control"/>
20             </div>
21             <div class="form-group">
22               <label for="kommentar">Kommentar</label>
23               <textarea id="kommentar" name="kommentar1" class="form-control">{{ old('kommentar1') }}</textarea>
24             </div>
25             <input type="submit" value="neuen Beitrag anlegen" class="btn btn-success mt-2"/>
26           </form>
27
28           <a href="/post" class="btn btn-primary mt-2 text-end float-end">
29             zurück
30           </a>
31         </div>
32       </div>
33     </div>
34   </div>
35 </div>
36 @endsection
```

{} old('name1') {}



MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Bitte überprüfe deine Eingaben

- Feld Kommentar1 ist ein Pflichtfeld!

Neuen Beitrag erstellen

Name

Tini

Kommentar

neuen Beitrag anlegen

zurück

3) Formular in DB eintragen

Post.php - Unbenannt (Arbeitsbereich) - Visual Studio Code

```
create.blade.php PostController.php Post.php
```

```
my_project_01 > app > Post.php
1 <?php
2
3 namespace App;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Post extends Model
8 {
9     #Die DBSpalten sind gesperrt und ich muss das damit erlauben
10    #mit $fillable erlaube ich es in diesen Spalten
11    protected $fillable = ['name', 'beschreibung'];
12 }
13
```

Beiträge verfassen | localhost / 127.0.0.1 / blog_laravel / +

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Neuen Beitrag erstellen

Name
Guten Morgen Welt

Kommentar
sfdfssdfsfsf

neuen Beitrag anlegen

zurück

localhost / 127.0.0.1 / blog_laravel / +

phpMyAdmin

Server: 127.0.0.1 > Datenbank: blog_laravel_webfull02 > Tabelle: posts

BJ...]

SELECT * FROM `posts` ORDER BY `beschreibung` ASC

Messen [Inline bearbeiten] [Bearbeiten] [SQL erklären] [PHP-Code erzeugen] [Aktualisieren]

Alles anzeigen Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sort

+ Optionen

	id	name	beschreibung	created_at	updated_at
<input type="checkbox"/>	1	Komme gleich raus	Der Lenz ist da	2023-06-28 08:17:53	2023-06-28 08:17:53
<input type="checkbox"/>	2	Frische Luft	consetetur sadipscing ...	NULL	NULL
<input type="checkbox"/>	3	Jeder	PHP ist einfach	NULL	NULL
<input type="checkbox"/>	4	Laravel ist einfach	consetetur sadipscing ...	NULL	NULL
<input type="checkbox"/>	5	Sonne tut gut	amet, consetetur sadipscing ...	NULL	NULL
<input type="checkbox"/>	6	Guten Morgen Welt	sfsdfssdfsfs	2023-06-28 08:20:03	2023-06-28 08:20:03

Alle auswählen markierte: Bearbeiten Kopieren Löschen Exportieren

Alles anzeigen Anzahl der Datensätze: 25 Zeilen filtern: Diese Tabelle durchsuchen Nach Schlüssel sort

Konsole

Drücken Sie Strg+Enter, um die Abfrage auszuführen

4) Formular in DB eintragen

-> Meldung, wenn falsche Eingabe anzeigen

PostController.php - Unbenannt (Arbeitsbereich) - Visual Studio Code

```
my_project_01 > app > Http > Controllers > PostController.php
```

```
41     * Store a newly created resource in storage.
42     * @param \Illuminate\Http\Request $request
43     * @return \Illuminate\Http\Response
44     */
45    public function store(Request $request)
46    {
47        //echo "ich bin ein Store";
48        [
49            'name1' => 'required|min:3',
50            'kommentar1' => 'required|min:5'
51        ]
52    }
53    #required = Pflichtfeld
54    #min:3 - mindestens 3 Zeichen lang
55
56
57    $post = new Post(
58        [
59            // 'nameDerSpalteInDB' => $request['nameDesFeldesInFormular']
60            'name' => $request['name1'],
61            'beschreibung' => $request['kommentar1']
62        ]
63    );
64    $post->save();
65    return redirect('/post');
```

app.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code

```
my_project_01 > resources > views > layouts > app.blade.php
```

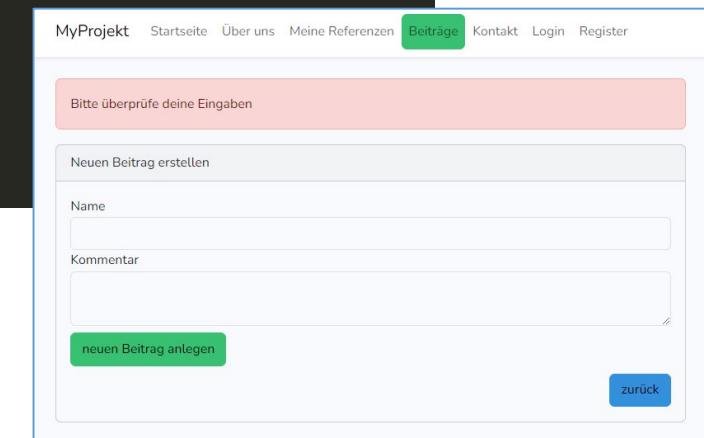
```
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
```

```
aria-labeledby="navbarDropdown">
<a class="dropdown-item" href="{{ route('logout') }}"
    onclick="event.preventDefault();
    document.getElementById('logout-form').submit();"
    {{ __('Logout') }}>
</a>

<form id="logout-form" action="{{ route('logout') }}" method="POST"
    class="d-none">
    @csrf
</form>
```

```
</div>
</li>
</ul>
</div>
</div>
</nav>

<main class="py-4">
@if($errors->any())
<div class="container">
    <div class="alert alert-danger">
        Bitte überprüfe deine Eingaben
    <ul>
        @foreach($errors->all() AS $fehler)
            <li>{!! $fehler !!}
```



Hinweise unter den Eingabefeldern anzeigen

<small class="form-text text-danger">{!! \$errors->first('name1') !!}</small>

```
index.blade.php create.blade.php
my_project_01 > resources > views > post > create.blade.php
1  @extends('layouts.app')
2
3  @section('title','Beiträge verfassen')
4
5  @section('content')
6  <div class="container">
7      <div class="row">
8          <div class="col-md-12">
9              <div class="card">
10                  <div class="card-header">
11                      Neuen Beitrag erstellen
12                  </div>
13                  <div class="card-body">
14
15                      <form action="/post" method="POST">
16                          @csrf
17                          <div class="form-group">
18                              <label for="name">Name</label>
19                              <input type="text" id="name" name="name1" value="{{ old('name1') }}" class="form-control"/>
20                              <small class="form-text text-danger">{!! $errors->first('name1') !!}</small>
21                              <!-- ->all() alle anzeigen und any() irgendeine anzeigen-->
22                          </div>
23                          <div class="form-group">
24                              <label for="kommentar">Kommentar</label>
25                              <textarea id="kommentar" name="kommentar1" class="form-control">{{ old('kommentar1') }}</textarea>
26                              <small class="form-text text-danger">{!! $errors->first('kommentar1') !!}</small>
27                          </div>
28                          <input type="submit" value="neuen Beitrag anlegen" class="btn btn-success mt-2"/>
29
30
31                      <a href="/post" class="btn btn-primary mt-2 text-end float-end">
32                          zurück
33                      </a>
34                  </div>
35              </div>
36          </div>
37      </div>
38  </div>
39  @endsection
```

The screenshots show a user interface for creating a new post. The top screenshot shows the error message in the code editor. The middle screenshot shows the error message displayed above the input fields. The bottom two screenshots show the error messages displayed below the input fields.

Screenshot 1: Error message above input fields

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Bitte überprüfe deine Eingaben

- Feld **Name1** ist ein Pflichtfeld!
- Feld **Kommentar1** ist ein Pflichtfeld!

Neuen Beitrag erstellen

Name

Feld **Name1** ist ein Pflichtfeld!

Kommentar

Feld **Kommentar1** ist ein Pflichtfeld!

neuen Beitrag anlegen

Screenshot 2: Error message below input fields (Name)

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Bitte überprüfe deine Eingaben

- Name1** muss mindestens 3 Zeichen haben.
- Kommentar1** muss mindestens 5 Zeichen haben.

Neuen Beitrag erstellen

Name

r

Name1 muss mindestens 3 Zeichen haben.

Kommentar

r

Kommentar1 muss mindestens 5 Zeichen haben.

neuen Beitrag anlegen

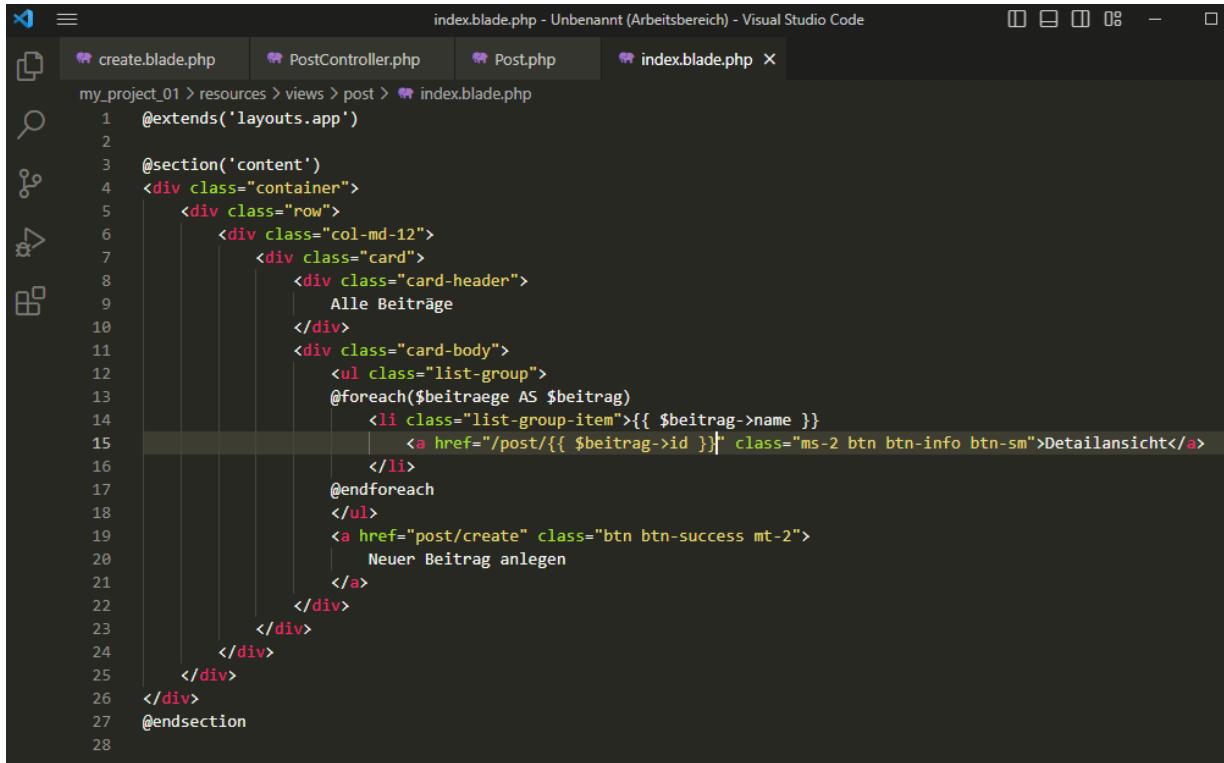
Screenshot 3: Error message below input fields (Kommentar)

zurück

Daten aus
DB einzeln anzeigen

1) DB Einträge einzeln anzeigen

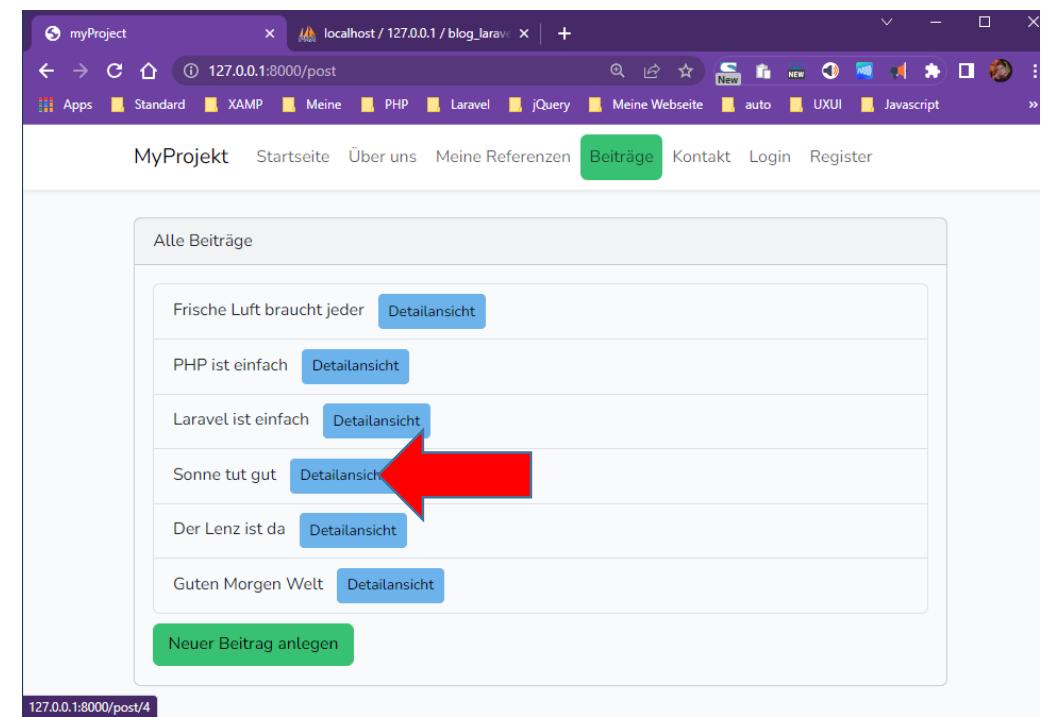
-> Hauptseite link einfügen



```
index.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code
create.blade.php PostController.php Post.php index.blade.php

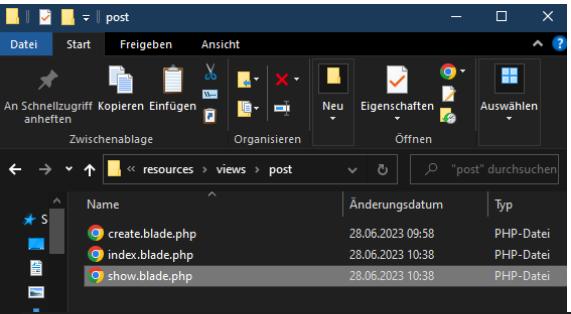
my_project_01 > resources > views > post > index.blade.php

1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-12">
7       <div class="card">
8         <div class="card-header"> Alle Beiträge
9       </div>
10      <div class="card-body">
11        <ul class="list-group">
12          @foreach($beitraege AS $beitrag)
13            <li class="list-group-item">{{ $beitrag->name }}
14              <a href="/post/{{ $beitrag->id }}" class="ms-2 btn btn-info btn-sm">Detailansicht</a>
15            </li>
16          @endforeach
17        </ul>
18        <a href="post/create" class="btn btn-success mt-2">
19          Neuer Beitrag anlegen
20        </a>
21      </div>
22    </div>
23  </div>
24 </div>
25 </div>
26 </div>
27 @endsection
```



2) DB Einträge einzeln anzeigen

→ showseite anlegen



```
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-12">
7       <div class="card">
8         <div class="card-header">
9           Beitrag - Detailansicht
10        </div>
11        <div class="card-body">
12          <p><b>{{ $post->name }}</b></p>
13          <p>{{ $post->beschreibung }}</p>
14          <a href="/post" class="btn btn-primary mt-2 float-end">
15            zurück
16          </a>
17        </div>
18      </div>
19    </div>
20  </div>
21 </div>
22 @endsection
```

```
59
60
61 }
62
63 /**
64 * Display the specified resource.
65 *
66 * @param \App\Post $post
67 * @return Illuminate\Http\Response
68 */
69 public function show(Post $post)
70 {
71   //echo "ich bin show";
72
73   #geht in post/show.blade.php und
74   #übergibt in post die Variablen aus von $post
75   #(meine Variable, von Laravel zur Verfügung gestellte Variable)
76   return view('post.show')->with('post',$post);
77
78 /**
79 * Show the form for editing the specified resource.
80 */
81
```

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Beitrag - Detailansicht

Laravel ist einfach

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

[zurück](#)

3) DB Einträge einzeln anzeigen

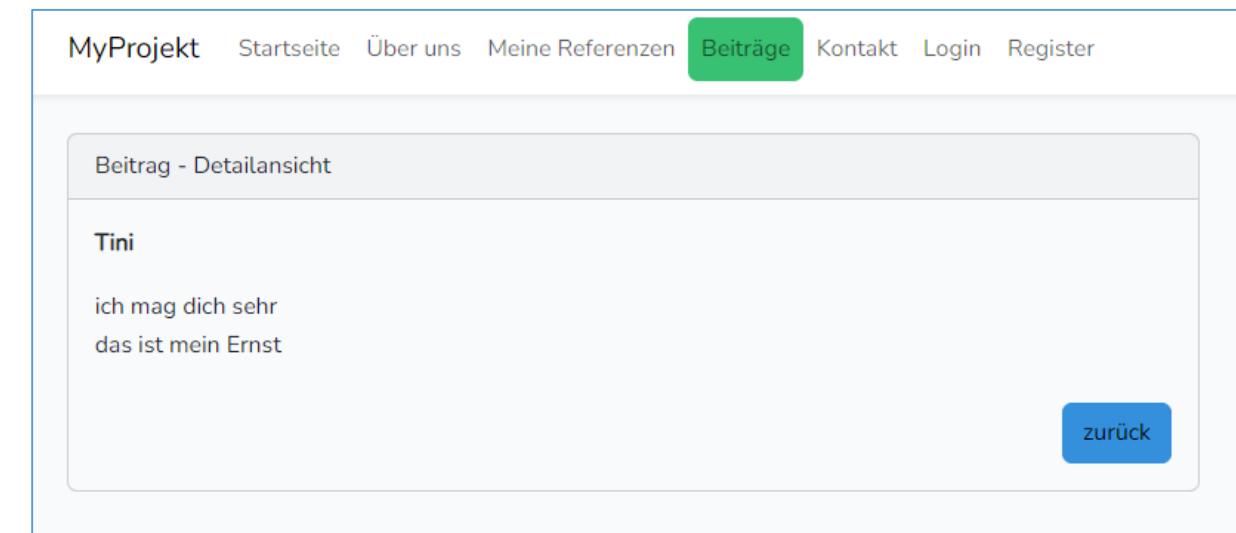
→ showseite anlegen mit {!! Xxx !!}

→ Damit hinterlegte HTML angezeigt wird

show.blade.php - Unbenannt (Arbeitsbereich) - Visual Studio Code

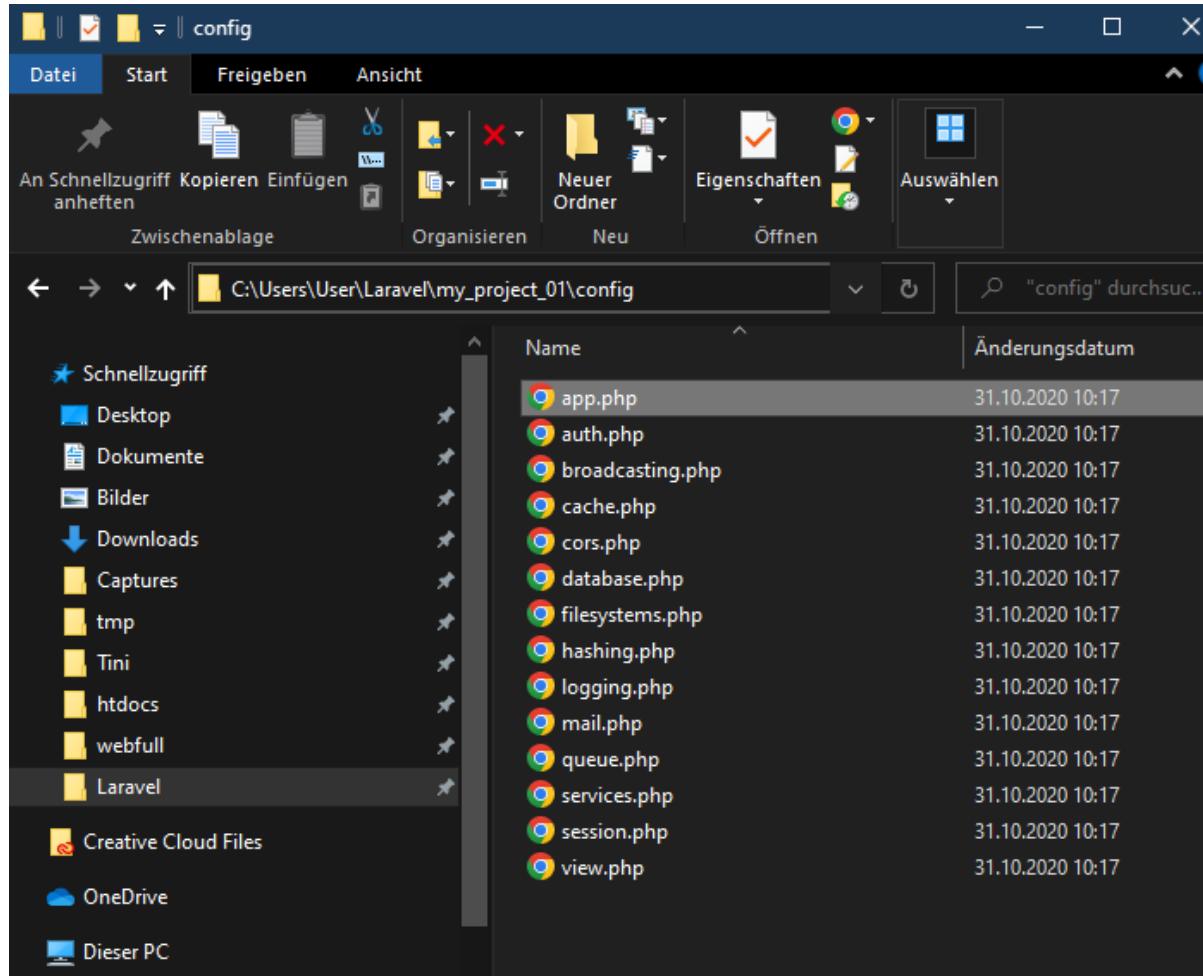
my_project_01 > resources > views > post > show.blade.php

```
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5   <div class="row">
6     <div class="col-md-12">
7       <div class="card">
8         <div class="card-header">
9           Beitrag - Detailansicht
10        </div>
11        <div class="card-body">
12          <p><b>{{ $post->name }}</b></p>
13          <!-- nl2br - new line 2 break <br> einfügen -->
14          <!-- 1xKlammer und 2x!! - Inhalt wird als HTML dargestellt-->
15          <p>{!! nl2br($post->beschreibung) !!}</p>
16          <a href="/post" class="btn btn-primary mt-2 float-end">
17            zurück
18          </a>
19        </div>
20      </div>
21    </div>
22  </div>
23 @endsection
```



Sprache ändern

1) Sprache ändern englisch in deutsch

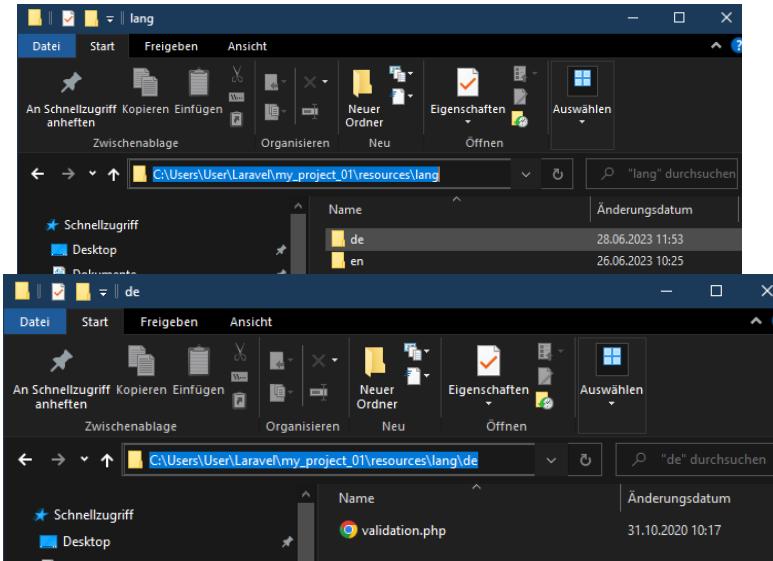


A screenshot of Visual Studio Code showing the "app.php" file in the "my_project_01 > config" folder. The code defines application configuration settings:

```
my_project_01 > config > app.php
-----
|----- Application Timezone -----
| Here you may specify the default timezone for your application, which
| will be used by the PHP date and date-time functions. We have gone
| ahead and set this to a sensible default for you out of the box.
|
|----- Application Locale Configuration -----
|
| The application locale determines the default locale that will be used
| by the translation service provider. You are free to set this value
| to any of the locales which will be supported by the application.
|
|----- Application Fallback Locale -----
|
|----- Application Locale -----
|----- Application Fallback Locale -----
```

The code includes comments explaining the purpose of each section: Application Timezone, Application Locale Configuration, and Application Fallback Locale.

2) Sprache ändern englisch in deutsch



The screenshot illustrates a developer's workflow for translating validation messages from English to German. It includes:

- Visual Studio Code Editor:** Shows the 'validation.php' file with code like: 'string' => '<b style="text-transform: capitalize;">:attribute muss mindestens :min Zeichen haben.', and 'array' => 'The :attribute must have at least :min items.'.
- Terminal:** Displays the command 'artisan config:cache' being run, followed by the output: 'Configuration cache cleared!' and 'Configuration cached successfully!'. The path 'C:\Users\User\ Laravel\my_project_01' is shown.
- Browser Preview:** A screenshot of a web page titled 'MyProjekt' with a navigation bar. The main content area contains a red box with the message: 'Bitte überprüfe deine Eingaben' and a bulleted list: '• Feld Name1 ist ein Pflichtfeld!' and '• Feld Kommentar1 ist ein Pflichtfeld!'. Below this is a form for creating a new post, with fields for 'Name' and 'Kommentar', and a green button labeled 'neuen Beitrag anlegen'.

Symbole einfügen

Symbole mit fa einfügen

```
index.blade.php X
my_project_01 > resources > views > post > index.blade.php
1 @extends('layouts.app')
2
3 @section('content')
4 <div class="container">
5     <div class="row">
6         <div class="col-md-12">
7             <div class="card">
8                 <div class="card-header">
9                     Alle Beiträge
10                </div>
11                <div class="card-body">
12                    <ul class="list-group">
13                        @foreach($beitraege AS $beitrag)
14                            <li class="list-group-item">
15                                <i class="fa-solid fa-comment text-warning"></i>
16                                {{ $beitrag->name }}
17                                <a href="/post/{{ $beitrag->id }}" class="ms-2 btn btn-info btn-sm">
18                                    <i class="fa fa-circle-info text-light"></i>
19                                    Detailansicht
20                                </a>
21                                <!--#####
22                                <a href="#" class="btn btn-outline-primary btn-sm">
23                                    <i class="fa fa-pen-to-square"></i>
24                                    Bearbeiten
25                                </a>
26                            </li>
27                        @endforeach
28                    </ul>
29                    <a href="post/create" class="btn btn-success mt-2">
30                        <i class="fa-regular fa-plus"></i>
31                        Neuer Beitrag anlegen
32                    </a>
33                </div>
34            </div>
35        </div>
36    </div>
37 </div>
38 @endsection
```

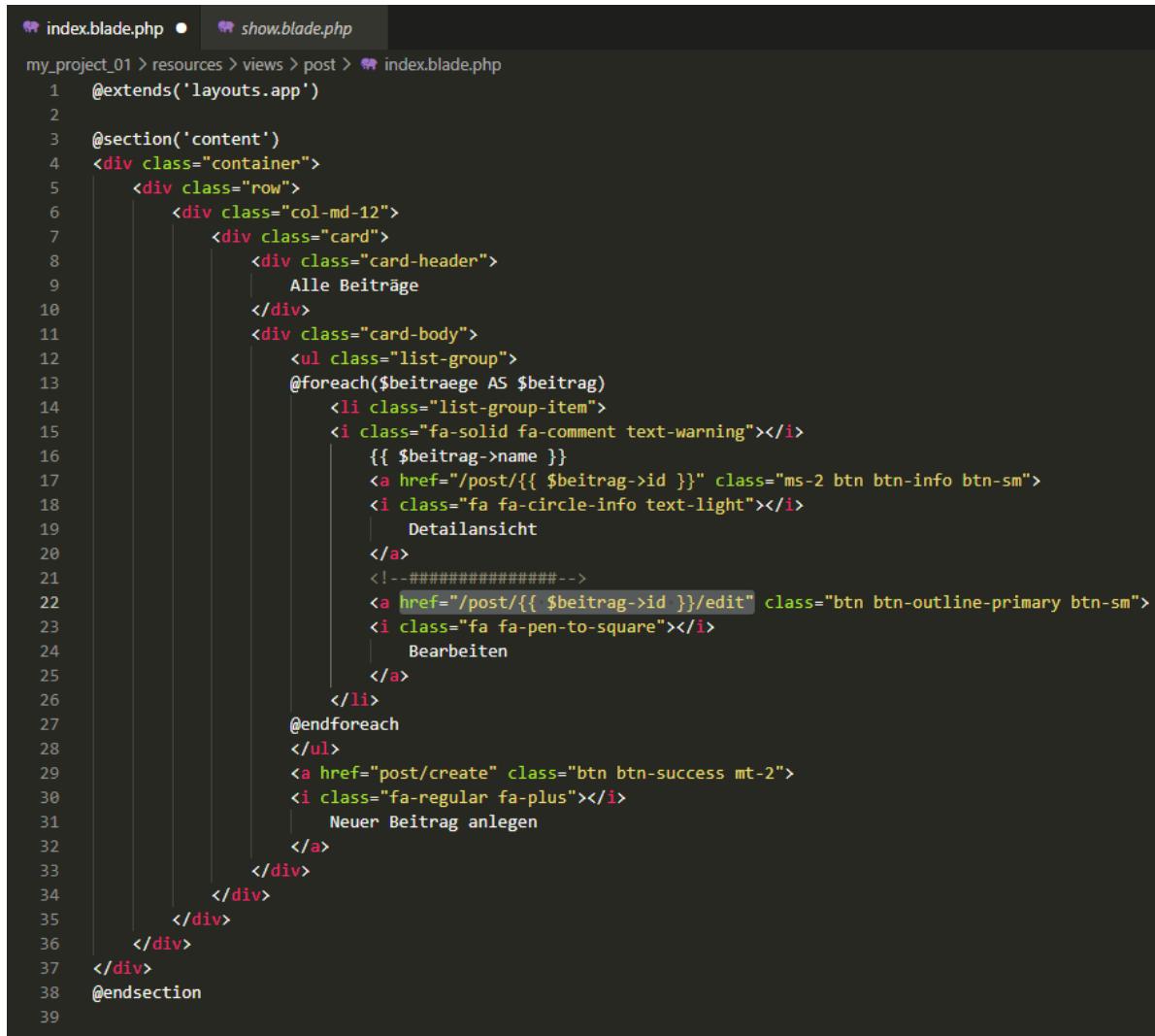
The screenshot shows a web application interface titled "MyProjekt". The top navigation bar includes links for "Startseite", "Über uns", "Meine Referenzen", a green highlighted "Beiträge", "Kontakt", "Login", and "Register". Below this, a section titled "Alle Beiträge" displays a list of posts. Each post entry consists of a yellow speech bubble icon, the post title, and two blue buttons labeled "Detailansicht" and "Bearbeiten". The posts listed are:

- Frische Luft braucht jeder
- PHP ist einfach
- Laravel ist einfach
- Sonne tut gut
- Der Lenz ist da
- Guten Morgen Welt
- Tini
- Klaus

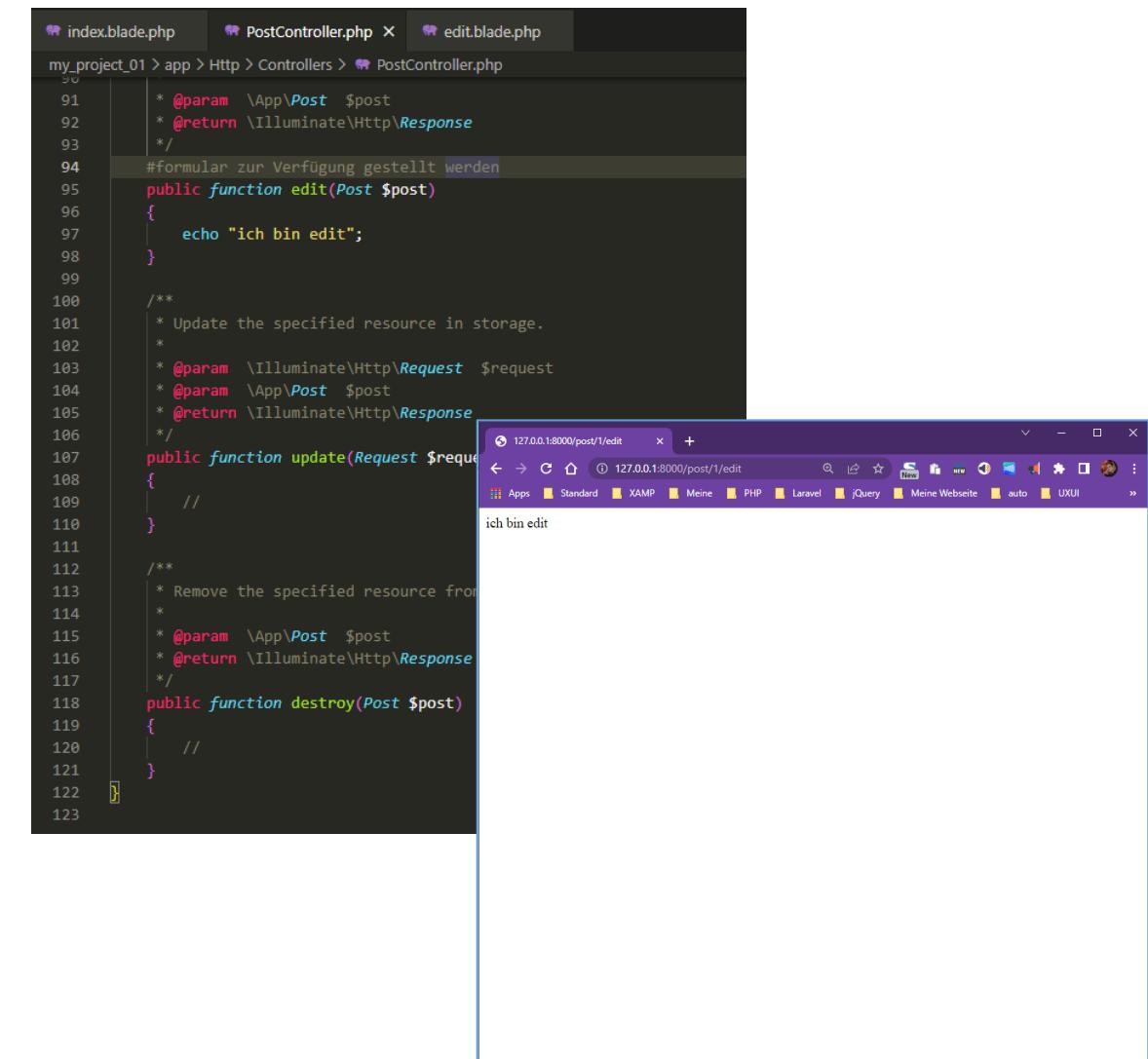
At the bottom of the list is a green button labeled "+ Neuer Beitrag anlegen".

Datensätze aus
DB bearbeiten

1) Datensätze bearbeiten -> Verbindung erstellen



```
index.blade.php
my_project_01 > resources > views > post > index.blade.php
1  @extends('layouts.app')
2
3  @section('content')
4      <div class="container">
5          <div class="row">
6              <div class="col-md-12">
7                  <div class="card">
8                      <div class="card-header">
9                          Alle Beiträge
10                     </div>
11                     <div class="card-body">
12                         <ul class="list-group">
13                             @foreach($beitraege AS $beitrag)
14                                 <li class="list-group-item">
15                                     <i class="fa-solid fa-comment text-warning"></i>
16                                     {{ $beitrag->name }}
17                                     <a href="/post/{{ $beitrag->id }}" class="ms-2 btn btn-info btn-sm">
18                                         <i class="fa fa-circle-info text-light"></i>
19                                         Detailansicht
20                                     </a>
21                                     <!-- ####-->
22                                     <a href="/post/{{ $beitrag->id }}/edit" class="btn btn-outline-primary btn-sm">
23                                         <i class="fa fa-pen-to-square"></i>
24                                         Bearbeiten
25                                     </a>
26                                 </li>
27                             @endforeach
28                         </ul>
29                         <a href="post/create" class="btn btn-success mt-2">
30                             <i class="fa-regular fa-plus"></i>
31                             Neuer Beitrag anlegen
32                         </a>
33                     </div>
34                 </div>
35             </div>
36         </div>
37     @endsection
38 
```

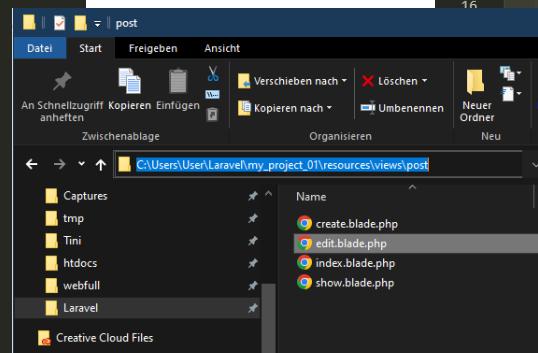


```
index.blade.php
PostController.php
edit.blade.php
my_project_01 > app > Http > Controllers > PostController.php
90
91     * @param \App\Post $post
92     * @return \Illuminate\Http\Response
93 */
94     #formular zur Verfügung gestellt werden
95     public function edit(Post $post)
96     {
97         echo "ich bin edit";
98     }
99
100 /**
101  * Update the specified resource in storage.
102  *
103  * @param \Illuminate\Http\Request $request
104  * @param \App\Post $post
105  * @return \Illuminate\Http\Response
106  */
107     public function update(Request $request)
108     {
109         //
110     }
111
112 /**
113  * Remove the specified resource from storage.
114  *
115  * @param \App\Post $post
116  * @return \Illuminate\Http\Response
117  */
118     public function destroy(Post $post)
119     {
120         //
121     }
122 }
123 
```

127.0.0.1:8000/post/1/edit

ich bin edit

2) Datensätze bearbeiten -> Datensatz aufrufen



The screenshot shows a Windows file explorer window titled "post" containing the following files:

- Captures
- tmp
- Tini
- htdocs
- webfull
- Laravel
- Creative Cloud Files
- index.blade.php
- create.blade.php
- edit.blade.php
- show.blade.php

Below the file explorer are two code editors side-by-side.

Left Editor (PostController.php):

```
index.blade.php PostController.php X edit.blade.php show.blade.php
my_project_01 > app > Http > Controllers > PostController.php

1 * @param \App\Post $post
2 * @return \Illuminate\Http\Response
3 */
4 #formular zur Verfügung gestellt werden
5 public function edit(Post $post)
6 {
7     //echo "ich bin edit";
8
9     return view('post.edit')->with('post',$post);
10 }
11
12 /**
13 * Update the specified resource in storage.
14 *
15 * @param \Illuminate\Http\Request $request
16 * @param \App\Post $post
17 * @return \Illuminate\Http\Response
18 */
19 public function update(Request $request, Post $post)
20 {
21     //
22 }
23
24 /**
25 * Remove the specified resource from storage.
26 *
27 * @param \App\Post $post
28 * @return \Illuminate\Http\Response
29 */
30 public function destroy(Post $post)
31 {
32     //
33 }
34
35 }
```

Right Editor (edit.blade.php):

```
index.blade.php PostController.php X edit.blade.php X create.blade.php
my_project_01 > resources > views > post > edit.blade.php
1 @extends('layouts.app')
2
3 @section('title','Beiträge bearbeiten')
4
5 @section('content')
6 <div class="container">
7     <div class="row">
8         <div class="col-md-12">
9             <div class="card">
10                 <div class="card-header">
11                     <i class="fa fa-solid fa-pen-to-square"></i>
12                     Beitrag bearbeiten
13                 </div>
14                 <div class="card-body">
15
16
17                 <form action="/post/{{ $post->id }}" method="POST">
18                     @csrf
19                     <!-- Laravel weis mit dem Befehl - muss in die Function Update reingegen-->
20                     @method('PUT')
21                     <div class="form-group">
22                         <label for="name">Name</label>
23                         <input type="text" id="name" name="name" value="{{ $post->name }}"
24                             class="form-control {{ $errors->has('name1') ? 'border-danger' : '' }}"/>
25                     <!-- class= ist Fehlermeldung für name1 vorhanden dann schreibe border-danger ansonsten mache nix -->
26
27                         <small class="form-text text-danger">{{ $errors->first('name1') }}</small>
28                     <!-- --all() alle anzeigen und any( ) irgendeine anzeigen-->
29                 </div>
30                 <div class="form-group">
31                     <label for="kommentar">Kommentar</label>
32                     <textarea id="kommentar" name="kommentar1"
33                         class="form-control {{ $errors->has('kommentar1') ? 'border-danger' : '' }}"><{{ $post->beschreibung }}>
34                     <!-- class= ist Fehlermeldung für kommentar1 vorhanden dann schreibe border-danger ansonsten mache nix -->
35                     <small class="form-text text-danger">{{ $errors->first('kommentar1') }}</small>
36                 </div>
37                 <button type="submit" class="btn btn-success mt-2"><i class="fa-solid fa-floppy-disk"></i> Beitrag ändern und speichern</button>
38             </form>
39
40             <a href="/post" class="btn btn-primary mt-2 text-end float-end">
41                 zurück
42             </a>
43         </div>
44     </div>
45   </div>
46 </div>
47 @endsection
```

3) Datensätze bearbeiten -> Änderung speichern

The screenshot illustrates the workflow for editing a post in a Laravel application:

- Code Editor:** Shows the `index.blade.php`, `PostController.php`, `edit.blade.php`, and `create.blade.php` files.
- Browser:** Shows the URL `127.0.0.1:8000/post/7`. The page displays the message "ich bin update".
- Form:** The `edit.blade.php` view contains a form for updating a post. It includes fields for Name and Kommentar, with validation messages for minimum character requirements.
- Console:** The `PostController.php` file shows the `update` method, which validates the input and updates the database using the `update` method on the `Post` model.
- Success:** The final step shows the updated post details: "TiniLi" and "Ich mag dich sehr! Das ist mein Ernst!" with a success message "Beitrag ändert und speichern".
- Detail View:** A separate view shows the detailed post information: "TiniLi" and "Ich mag dich sehr! Das ist mein Ernst!".

4) Datensätze bearbeiten -> Anzeige Daten aus DB, wenn leere Eingabe

```
index.blade.php PostController.php edit.blade.php create.blade.php

my_project_01 > resources > views > post > edit.blade.php
1  @extends('layouts.app')
2
3  @section('title','Beiträge bearbeiten')
4
5  @section('content')
6      <div class="container">
7          <div class="row">
8              <div class="col-md-12">
9                  <div class="card">
10                     <div class="card-header">
11                         <i class="fa fa-solid fa-pen-to-square"></i>
12                         Beitrag bearbeiten
13                     </div>
14                     <div class="card-body">
15
16                         <form action="/post/{{ $post->id }}" method="POST">
17                             @csrf
18                             <!-- Laravel weis mit dem Befehl - muss in die Function Update reingegen--&gt;
19                             @method('PUT')
20                             &lt;div class="form-group"&gt;
21                                 &lt;label for="name"&gt;Name&lt;/label&gt;
22                                 &lt;input type="text" id="name" name="name1" value="{{ old('name1') ?? $post-&gt;name }} " 
23                                     class="form-control {{ $errors-&gt;has('name1') ? 'border-danger' : '' }}"/&gt;
24                             <!-- Klasse ist Fehlermeldung für name1 vorhanden dann schreibe border-danger ansonsten mache nix --&gt;
25                                 &lt;small class="form-text text-danger"&gt;{!! $errors-&gt;first('name1') !!}&lt;/small&gt;
26                             <!-- &gt;all() alle anzeigen und any() irgendeine anzeigen--&gt;
27                         &lt;/div&gt;
28                         &lt;div class="form-group"&gt;
29                             &lt;label for="kommentar"&gt;Kommentar&lt;/label&gt;
30                             &lt;textarea id="kommentar" name="kommentar1"
31                                 class="form-control {{ $errors-&gt;has('kommentar1') ? 'border-danger' : '' }}&gt;{{ old('kommentar1') ?? $post-&gt;beschreibung }}&lt;/textarea&gt;
32                             <!-- Klasse ist Fehlermeldung für kommentar1 vorhanden dann schreibe border-danger ansonsten mache nix --&gt;
33                             &lt;small class="form-text text-danger"&gt;{!! $errors-&gt;first('kommentar1') !!}&lt;/small&gt;
34                         &lt;/div&gt;
35                         &lt;button type="submit" class="btn btn-success mt-2"&gt;&lt;i class="fa-solid fa-floppy-disk"&gt;&lt;/i&gt; Beitrag ändern und speichern&lt;/button&gt;
36
37                         &lt;a href="/post" class="btn btn-primary mt-2 text-end float-end"&gt;
38                             zurück
39                         &lt;/a&gt;
40                     &lt;/div&gt;
41                 &lt;/div&gt;
42             &lt;/div&gt;
43         &lt;/div&gt;
44     &lt;/div&gt;
45     @endsection</pre>
```

The screenshot displays a sequence of five application screenshots illustrating the validation process for an empty input field. Each screenshot shows a 'Beitrag bearbeiten' (Post Edit) form with two fields: 'Name' and 'Kommentar'. A green 'Beitrag ändern und speichern' (Change and Save) button is present in each.

- Screenshot 1:** Both fields ('Name' and 'Kommentar') contain valid data: 'Der Lenz ist da' and 'Komme gleich raus'. The button is green.
- Screenshot 2:** The 'Name' field is empty, while 'Kommentar' contains 'Komme gleich raus'. The button is green.
- Screenshot 3:** Both fields are empty. A red error message box appears, stating: 'Bitte überprüfe deine Eingaben' with bullet points: '• Feld Name1 ist ein Pflichtfeld!' and '• Feld Kommentar1 ist ein Pflichtfeld!'. The button is green.
- Screenshot 4:** The 'Name' field contains 'Laravel ist einfach' and the 'Kommentar' field contains placeholder text 'Lorem ipsum dolor sit amet, consetetur sadipscing elitr, nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam'. The button is green.
- Screenshot 5:** The 'Name' field contains 'La' and the 'Kommentar' field contains placeholder text. A red error message box appears, stating: 'Bitte überprüfe deine Eingaben' with bullet points: '• Name1 muss mindestens 3 Zeichen haben.' The button is green.

Datensätze aus
DB löschen

1) Datensatz löschen... Button anlegen

```
index.blade.php ●
my_project_01 > resources > views > post > index.blade.php
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

```
<a href="/post/{{ $beitrag->id }}" class="ms-2 btn btn-info btn-sm">
    <i class="fa fa-circle-info text-light"></i>
    Detailansicht
</a>
<!--#####
-->
<a href="/post/{{ $beitrag->id }}/edit" class="btn btn-outline-primary btn-sm">
    <i class="fa fa-pen-to-square"></i>
    Bearbeiten
</a>
<!--#-->

<form action="/post/{{ $beitrag->id }}" method="POST" class="d-inline-block">
    @csrf
    @method('DELETE')
    <button type="submit" class="btn btn-outline-danger btn-sm"><i class="fa fa-trash"></i> Löschen</button>
</form>

</li>
@endforeach
</ul>
<a href="post/create" class="btn btn-success mt-2">
    <i class="fa-regular fa-plus"></i>
    Neuer Beitrag anlegen
</a>
</div>
</div>
</div>
</div>
@endsection
```

Alle Beiträge

Frische Luft braucht jeder	Detailansicht	Bearbeiten	Löschen
PHP ist einfach	Detailansicht	Bearbeiten	Löschen
Laravel ist einfach	Detailansicht	Bearbeiten	Löschen
Sonne tut gut	Detailansicht	Bearbeiten	Löschen
Der Lenz ist da	Detailansicht	Bearbeiten	Löschen
Guten Morgen Welt	Detailansicht	Bearbeiten	Löschen
TiniLi	Detailansicht	Bearbeiten	Löschen
Klaus	Detailansicht	Bearbeiten	Löschen

+ Neuer Beitrag anlegen

2) Datensatz löschen ...

```
index.blade.php
my_project_01 > app > Http > Controllers > PostController.php
127.0.0.1:8000/post/5
127.0.0.1:8000/post/5

PostController.php
my_project_01 > app > Http > Controllers > PostController.php
127.0.0.1:8000/post
127.0.0.1:8000/post

    name => $request->name1,
    'beschreibung' => $request->kommentar1
);

return redirect('/post');

}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Post $post
 * @return \Illuminate\Http\Response
 */
public function destroy(Post $post)
{
    echo "ich bin der Löscher";
}
```

127.0.0.1:8000/post/5

localhost /

Apps Standard XAMP Meine

ich bin der Löscher

```
index.blade.php
my_project_01 > app > Http > Controllers > PostController.php
127.0.0.1:8000/post
127.0.0.1:8000/post

PostController.php
my_project_01 > app > Http > Controllers > PostController.php
127.0.0.1:8000/post
127.0.0.1:8000/post

    name => $request->name1,
    'beschreibung' => $request->kommentar1
);

return redirect('/post');

}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Post $post
 * @return \Illuminate\Http\Response
 */
public function destroy(Post $post)
{
    //echo "ich bin der Löscher";

    $post->delete();
    return redirect('/post');
}
```

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Alle Beiträge

Frische Luft braucht jeder	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
PHP ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Laravel ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Sonne tut gut	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Guten Morgen Welt	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
TiniLi	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>

+ Neuer Beitrag anlegen

myProject Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Auf 127.0.0.1:8000 wird Folgendes angezeigt:
Wollen Sie den Datensatz wirklich löschen?

OK Abbrechen

Alle Beiträge

Frische Luft braucht jeder	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
PHP ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Laravel ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Sonne tut gut	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Guten Morgen Welt	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
TiniLi	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>

+ Neuer Beitrag anlegen

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Frische Luft braucht jeder	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
PHP ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Laravel ist einfach	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Sonne tut gut	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
Guten Morgen Welt	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>
TiniLi	<input type="button" value="Detailansicht"/>	<input type="button" value="Bearbeiten"/>	<input type="button" value="Löschen"/>

+ Neuer Beitrag anlegen

Daten aus DB anzeigen
z.B. Datum

Datum wann angelegt anzeigen

```
index.blade.php x PostController.php
my_project_01 > resources > views > post > index.blade.php
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
</a>
<!--#####



Bearbeiten
</a>
<!--#####
<form action="/post/{{ $beitrag->id }}" method="POST" class="d-inline-block"
onclick="return confirm('Wollen Sie den Datensatz wirklich löschen');">
@csrf
@method('DELETE')
<button type="submit" class="btn btn-outline-danger btn-sm"><i class="fa fa-trash"></i> Löschen</button>
</form>
<!--#####
{{ $beitrag->created_at}}

```

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Alle Beiträge

Guten Morgen Welt	Detailansicht	Bearbeiten	Löschen	2023-06-28 08:20:03
TiniLi	Detailansicht	Bearbeiten	Löschen	2023-06-28 09:19:24
Guten Morgen Welt	Detailansicht	Bearbeiten	Löschen	2023-06-29 11:35:12

+ Neuer Beitrag anlegen

```
index.blade.php ● PostController.php
my_project_01 > resources > views > post > indexblade.php
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
```

```
</a>
<!--#####
<form action="/post/{{ $beitrag->id }}" method="POST" class="d-inline-block"
onclick="return confirm('Wollen Sie den Datensatz wirklich löschen');">
@csrf
@method('DELETE')
<button type="submit" class="btn btn-outline-danger btn-sm"><i class="fa fa-trash"></i> Löschen</button>
</form>
<!--#####
<div class="float-end">
@if ( $beitrag->created_at != NULL )
{
{{ $beitrag->created_at->diffForHumans() }}
}
@endif
</div>
</li>
@endforeach

```

MyProjekt Startseite Über uns Meine Referenzen Beiträge Kontakt Login Register

Alle Beiträge

Guten Morgen Welt	Detailansicht	Bearbeiten	Löschen	{ vor 1 Tag }
TiniLi	Detailansicht	Bearbeiten	Löschen	{ vor 1 Tag }
Guten Morgen Welt	Detailansicht	Bearbeiten	Löschen	{ vor 7 Minuten }

+ Neuer Beitrag anlegen