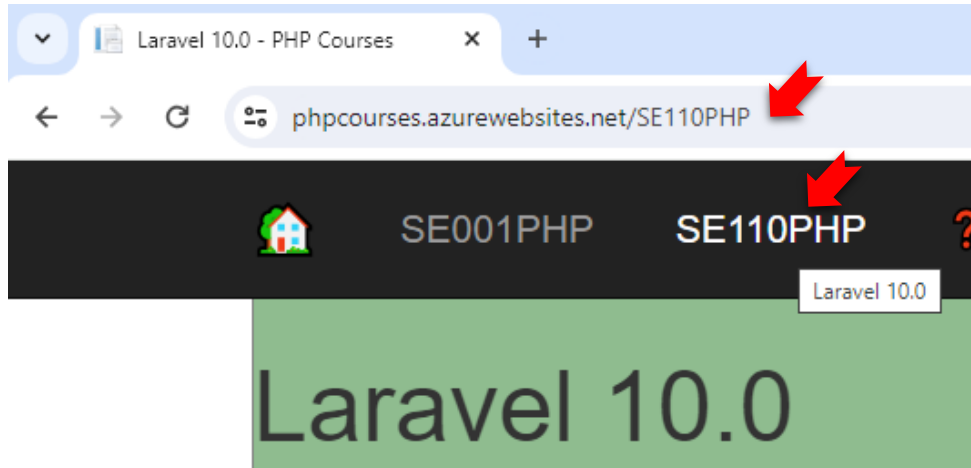


AJAX Mini-Project

Prepare course supporting materials

- 1) Download AJAX.Zip file from course supporting site

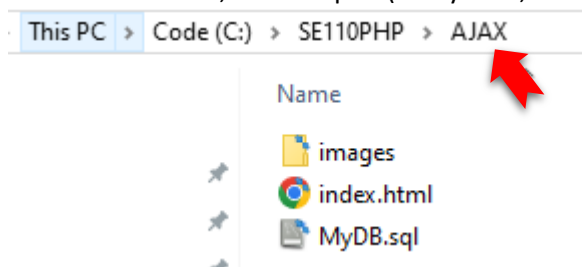


- 2) Scroll down until you see the download link

AJAX:

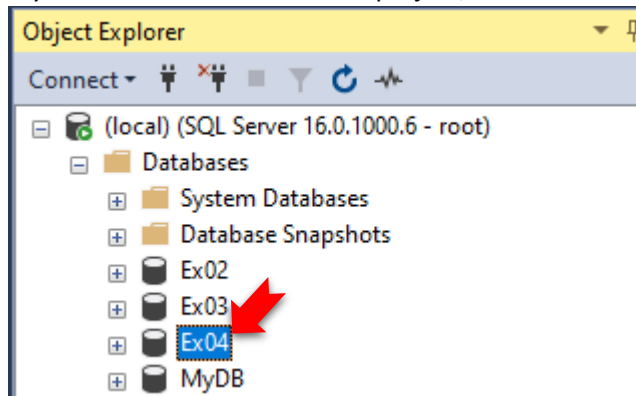
- [AJAX Exercise](#)
- [AJAX mini project exercise supporting material](#)
-  [How to Upload File in Laravel Using AJAX](#)

- 3) Click to download, and unzip it. (In my case, I unzipped it to C:\SE110PHP\)

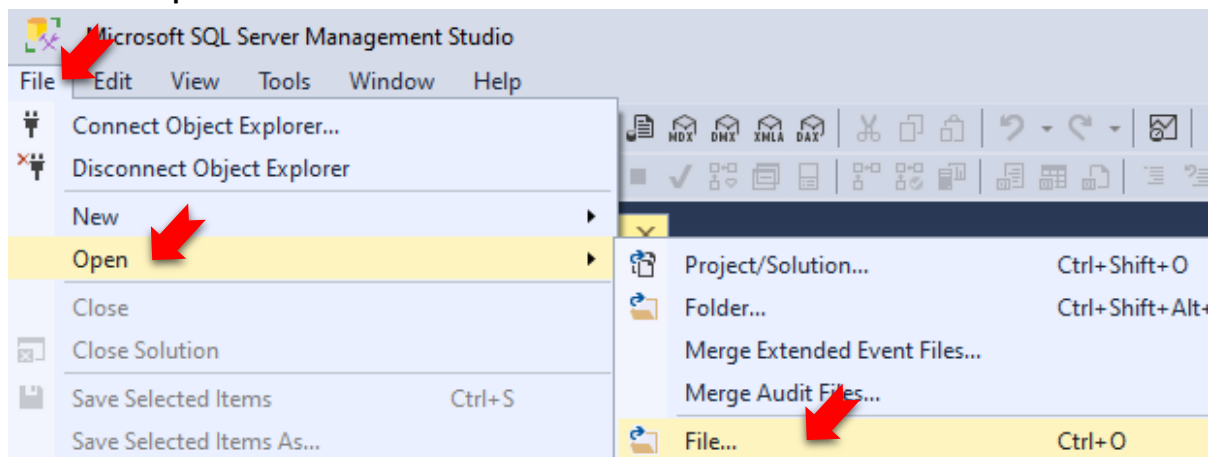


Prepare supporting tables for the mini project

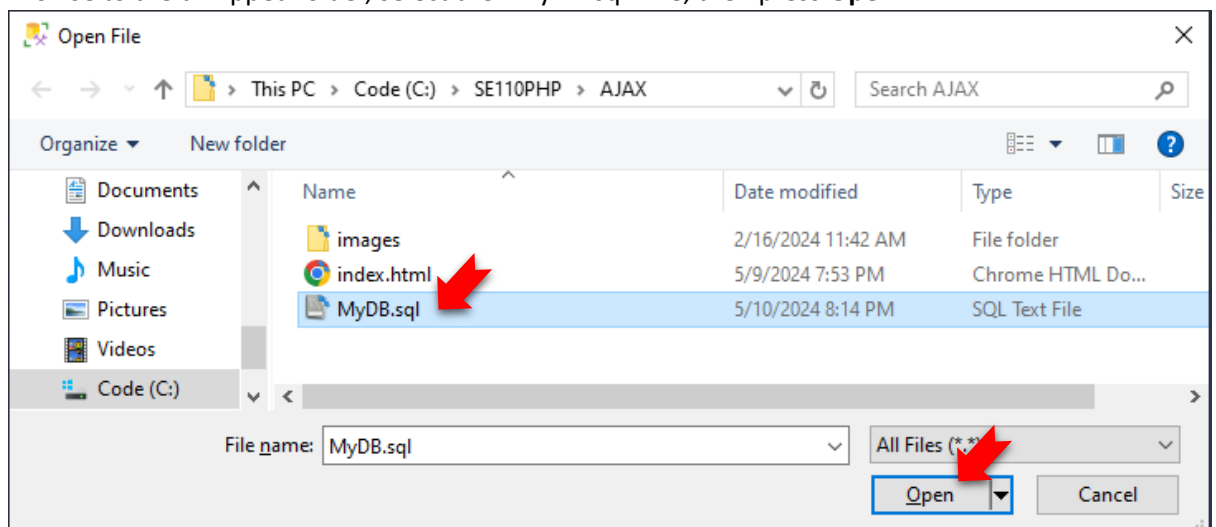
- 1) Start SSMS
- 2) Sign in using **root** account
- 3) If you don't have the database project, create a new one. (In my case, the database is **Ex04**)



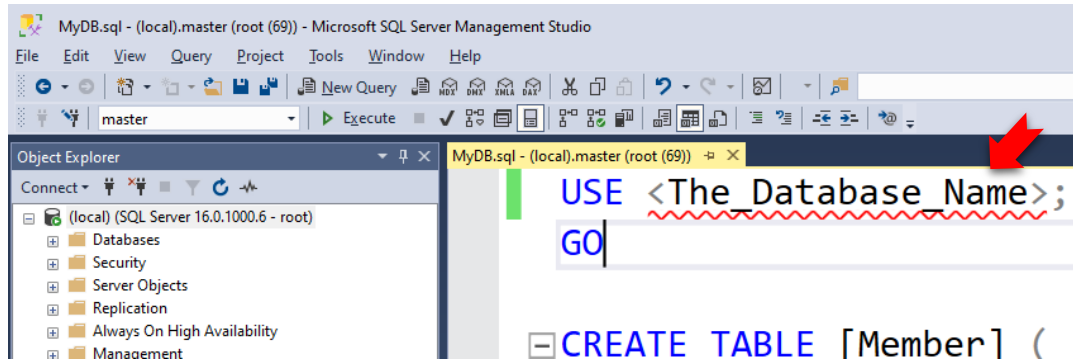
- 4) Select **File->Open->File...**



- 5) Browse to the unzipped folder, select the "MyDB.sql" file, then press **Open**



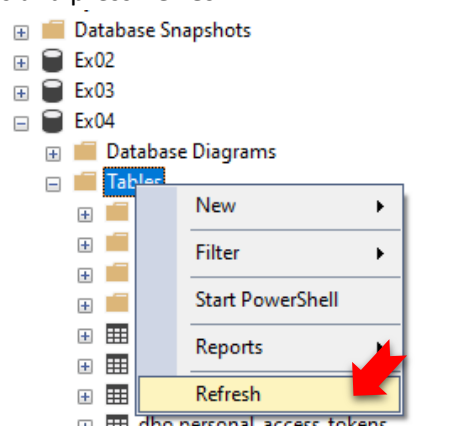
- 6) Replace the database name before running the script



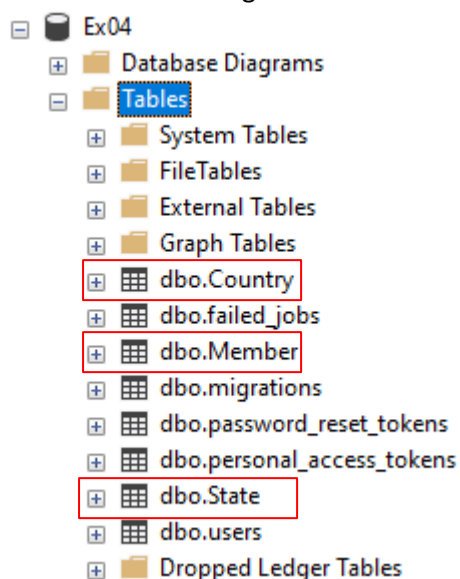
- 7) Make sure that don't select any statement in the script file, then press the **Execute** button



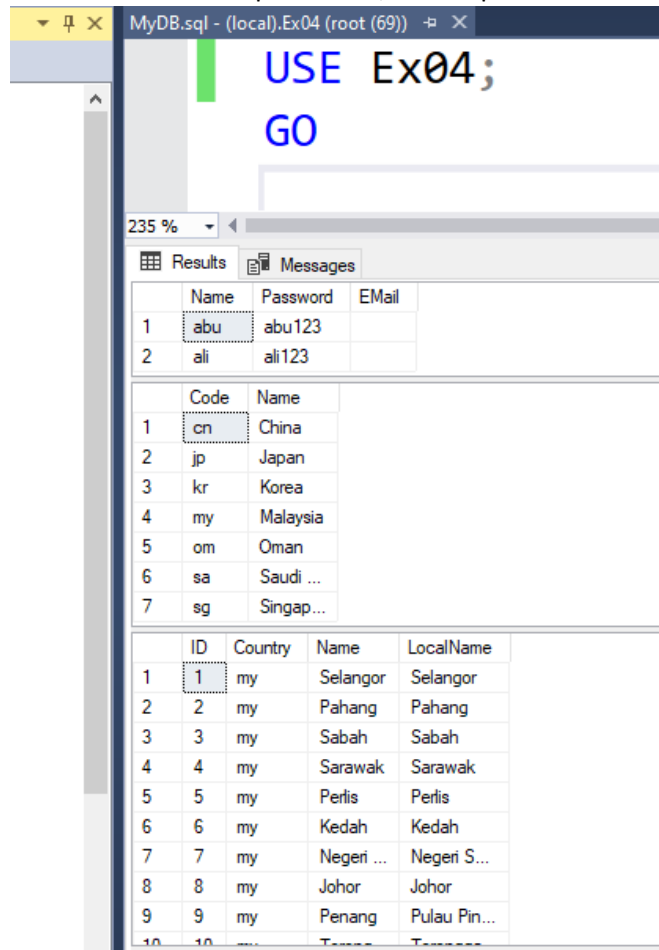
- 8) Assume that the database name is "Ex04". After running the script, right click the database **Tables** and press **Refresh**



- 9) There are 3 new tables generate



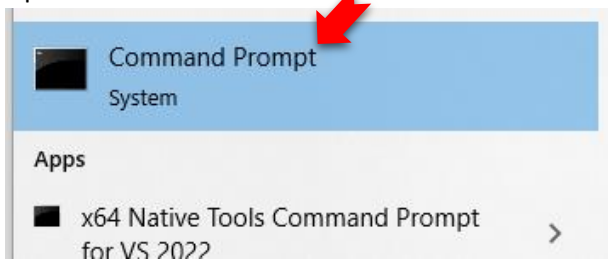
10) At the bottom of script window, the script also shows the following result



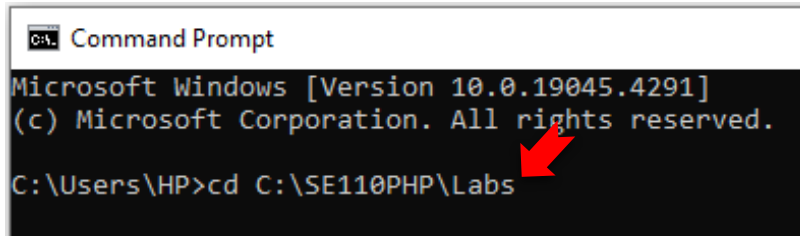
11) Close the Management Studio

Start a new Laravel 10 project

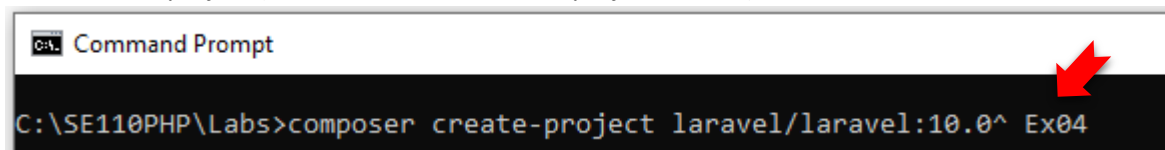
- 1) Open command window



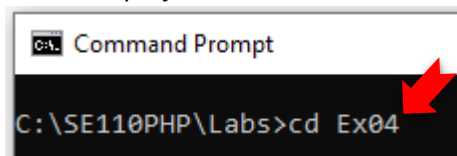
- 2) Browse to the labs folder (Here we assume is **C:\SE110PHP\Labs**)



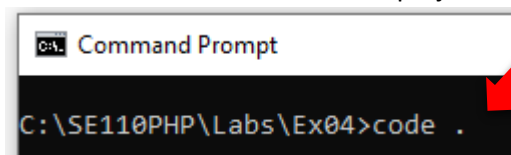
- 3) Create a new project (Here we assume that the project is **Ex04**)



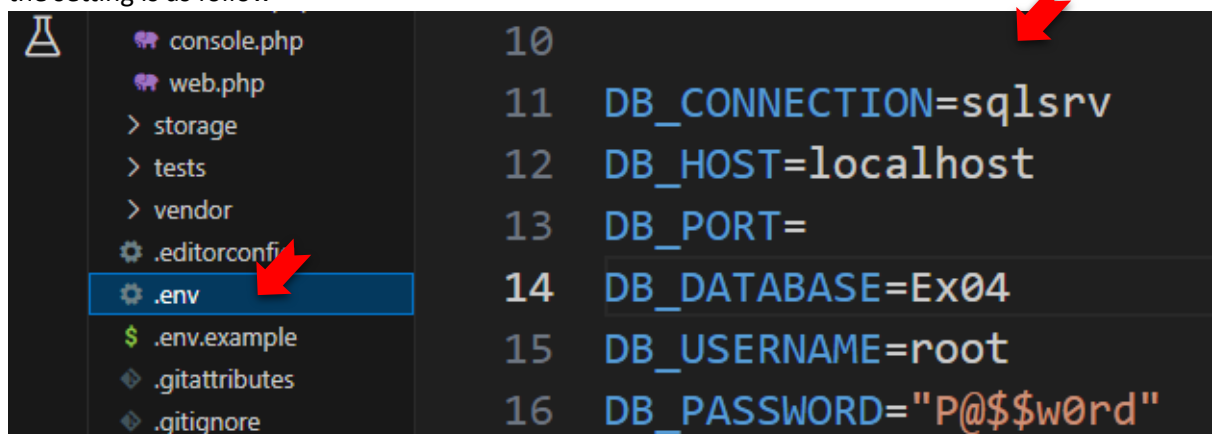
- 4) Go to the project folder



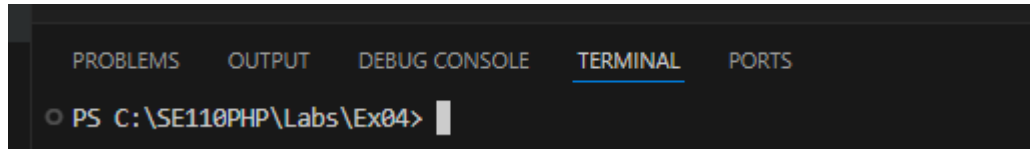
- 5) Run Visual Studio Code from the project folder



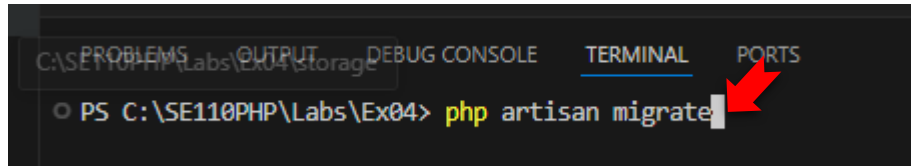
- 6) In the project explorer windows, select **.env** and scroll to the database section. Make sure the setting is as follow



- 7) Open the Terminal window



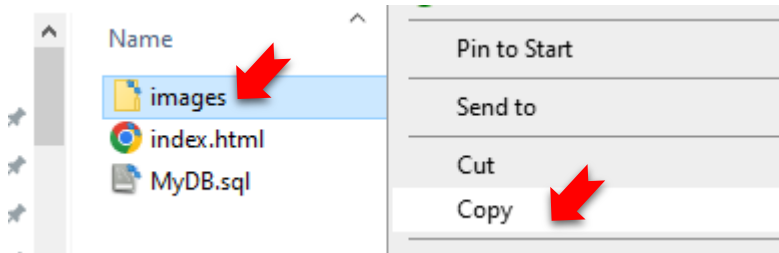
- 8) Run the migration



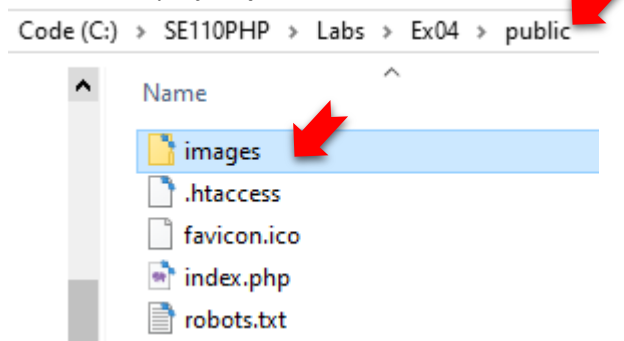
If no error, means the database setting is fine.

Copy the supporting images

- 1) From the unzipped folder, copy the entire images folder

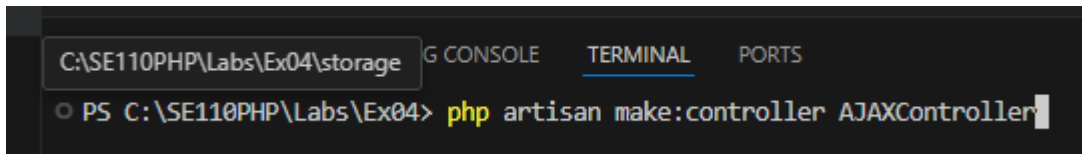


- 2) Paste to the project **public** folder



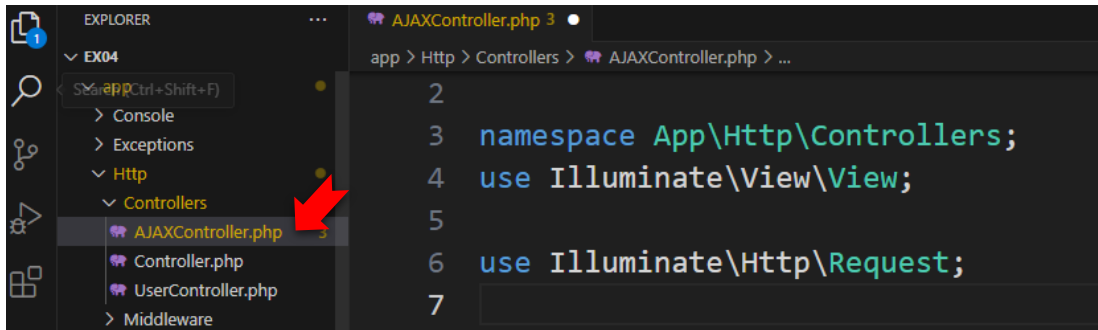
Create AJAX Controller

- 1) Now is time to add new **AJAX** controller



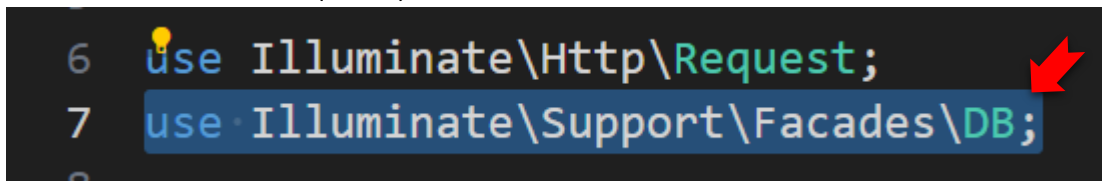
```
C:\SE110PHP\Labs\Ex04\storage G CONSOLE TERMINAL PORTS
PS C:\SE110PHP\Labs\Ex04> php artisan make:controller AJAXController
```

- 2) A new Controller file is created



```
2
3 namespace App\Http\Controllers;
4 use Illuminate\View\View;
5
6 use Illuminate\Http\Request;
7
```

- 3) Add the line of statement (line#7) in the controller file



```
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\DB;
```

- 4) Add the **Countries** function the AJAX controller class

```
public function Countries(Request $request):View{
    $result = DB::table('Country')->get();
    $countries = [];
    foreach ($result as $country) {
        array_push($countries,$country->Name);
    }
    return view('ajax.Countries', ['countries' => $countries]);
}
```

- 5) Add the **Countries** function the AJAX controller class

```
public function States(
    string $countryName,
    string $useLocalName = "no")
{
    $result = DB::select("SELECT s.[Name],s.[LocalName]
                        FROM State s
                        JOIN Country c ON s.Country=c.Code
                        WHERE c.Name=?",[$countryName]);

    $states = [];
    foreach ($result as $state) {
        array_push($states,($useLocalName=="yes")?
                    $state->LocalName:$state->Name);
    }
    if (count($states) == 0) return "";
    else return view('ajax.States', ['states'=>$states]);
}
```

- 6) Add the **CheckUser** function the AJAX controller class

```
public function CheckUser(string $userName): string{
    sleep(3); //Simulate connection latency
    return DB::scalar("SELECT COUNT(*)
                      FROM Member
                      WHERE [Name]=?",
                      [$userName])?"Found":"";
}
```

- 7) Add the **isLeapYear** supporting function the AJAX controller class

```
static private function isLeapYear(int $year):bool {
    if (($year % 4) != 0) return false;
    if (($year % 100) != 0) return true;
    if (($year % 400) != 0) return false;
    return true;
}
```

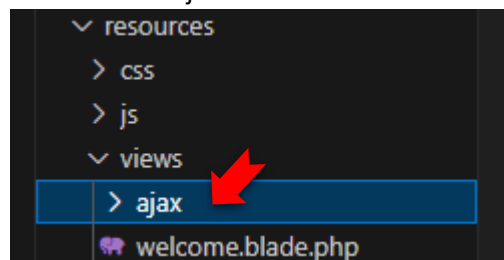
- 8) Add the **GetDays** function the AJAX controller class

```
public function GetDays(int $year, string $sMonth):string {
    $result = "<select id='Days' name='Days'>";
    $n = 31;
    switch ($sMonth) {
        case "APR":
        case "JUN":
        case "SEP":
        case "NOV":
            $n = 30;
            break;
        case "FEB":
            $n = $this::isLeapYear($year) ? 29 : 28;
            break;
    }
    for ($d = 1; $d <= $n; $d++) {
        $result = $result."<option>$d</option>";
    }
    return $result."</select>";
}
```

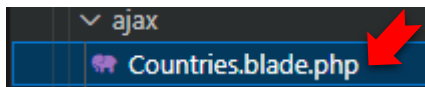
- 9) Save the AJAX control file

Create the needed view blade files

- 1) Create a new ajax view folder



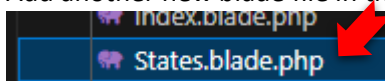
- 2) Add new blade file **Countries.blade.php**



The script:

```
<select id='Country' name='Country' onchange='loadStates()'>
  <option>--Select Country--</option>
  @foreach ($countries as $country)
    <option>{{$country}}</option>
  @endforeach
</select>
```

- 3) Add another new blade file in the ajax view folder **States.blade.php**

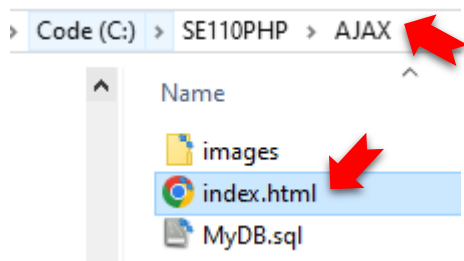


The script:

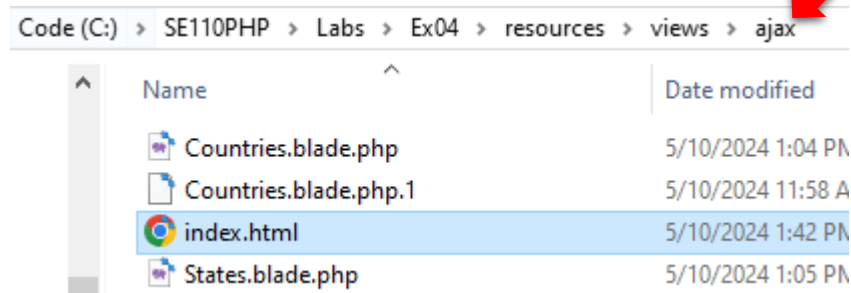
```
<select id='State' name='State'>
  @foreach ($states as $state)
    <option>{{$state}}</option>
  @endforeach
</select>
```

Prepare the AJAX index blade file

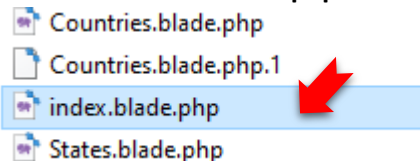
- 1) Copy the index.html file from the unzipped folder



- 2) Paste it to the AJAX controller view folder



- 3) Rename it as **index.blade.php**



- 4) Save All, now the project ready to go

Run and test the mini project

- 1) In the Terminal window, run the following command

```
PS C:\SE110PHP\Labs\Ex04> php artisan cache:clear
```

INFO Application cache cleared successfully.

- 2) Another command

```
PS C:\SE110PHP\Labs\Ex04> php artisan optimize
```

INFO Caching the framework bootstrap files.

config	46ms	DONE
routes	34ms	DONE

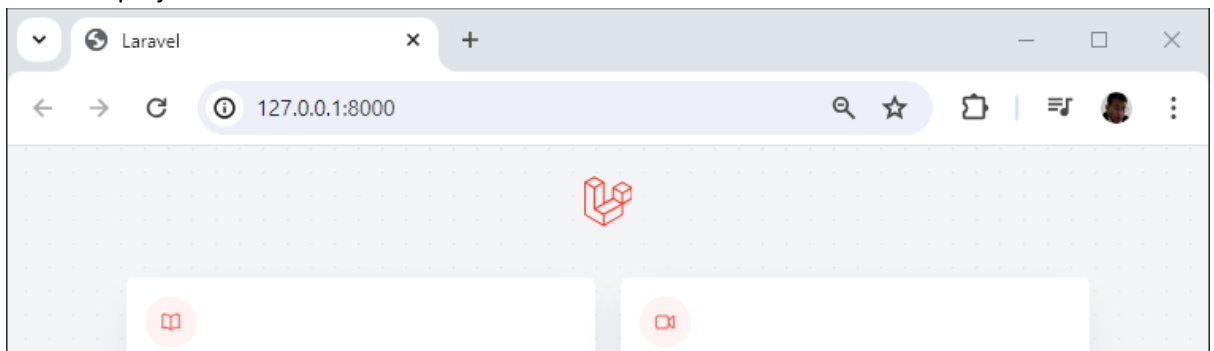
- 3) Start the server, press Ctrl-Click the url

```
PS C:\SE110PHP\Labs\Ex04> php artisan serve
```

INFO Server running on [http://127.0.0.1:8000].

Press Ctrl+C to stop the server

- 4) View the project from browser



- 5) Browse to <http://127.0.0.1:8000/AJAX>

A screenshot of a web browser window showing a 'Member Registration' form. The form is titled 'Member Registration' and contains several input fields: 'User:', 'Password:', 'Country:' (with a dropdown menu), 'State:', 'DOB:' (with year, month, and day dropdowns), and a 'Local Name' checkbox. A 'Submit' button is at the bottom. The browser's address bar shows '127.0.0.1:8000/AJAX'.

- 6) Key in **Ali** in the User textbox, press Tab, wait for 3 seconds

Member Registration	
User:	Ali  
Password:	

Note: Meaning that, the user Ali already exists. So, this user name is invalid

- 7) Now change to **Alibaba**, then Tab

Member Registration	
User:	Alibaba  
Password:	

Note: Meaning that, the user Alibaba is valid

- 8) Now try select Korea from the Country dropdown list

Country:	--Select Country-- 	Name	
State:	--Select Country--		
DOB:	China		1 
	Japan		
	Korea 		
	Malaysia		

Note: Observe what will happened to the State dropdown list below

Country:	Korea 	
State:	Busan 	☐ Local Name

- 9) Select the Local Name checkbox and observe the State dropdown list again

State:	부산시 	☒ Local Name 
--------	---	---

- 10) Lastly try to change the DOB Year and Month dropdown lists and try to find out how many days into the Day dropdown list

DOB:	2010 	JAN 	1  
------	--	---	---