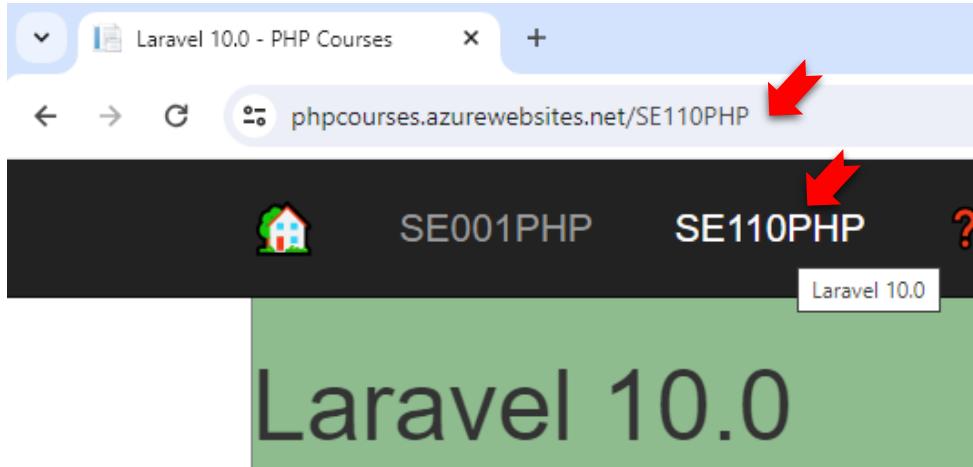


## AJAX Mini-Project

### Prepare course supporting materials

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

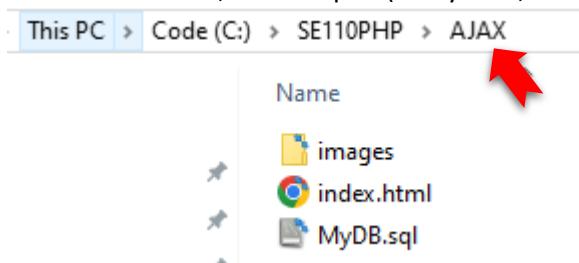


- 2) Scroll down until you see the download link

#### AJAX:

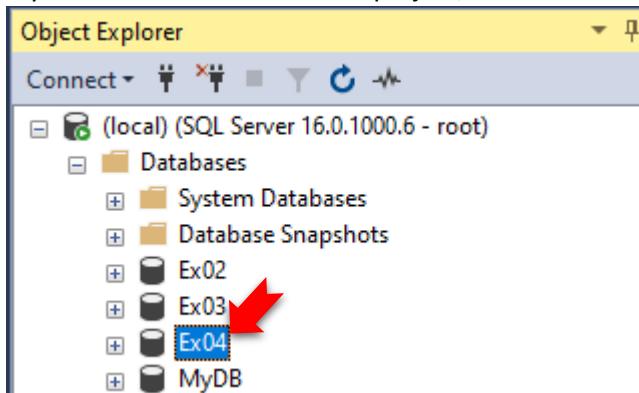
- [AJAX Exercise](#)
- [AJAX mini project exercise supporting material](#)
- [How to Upload File in Laravel Using AJAX](#)

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

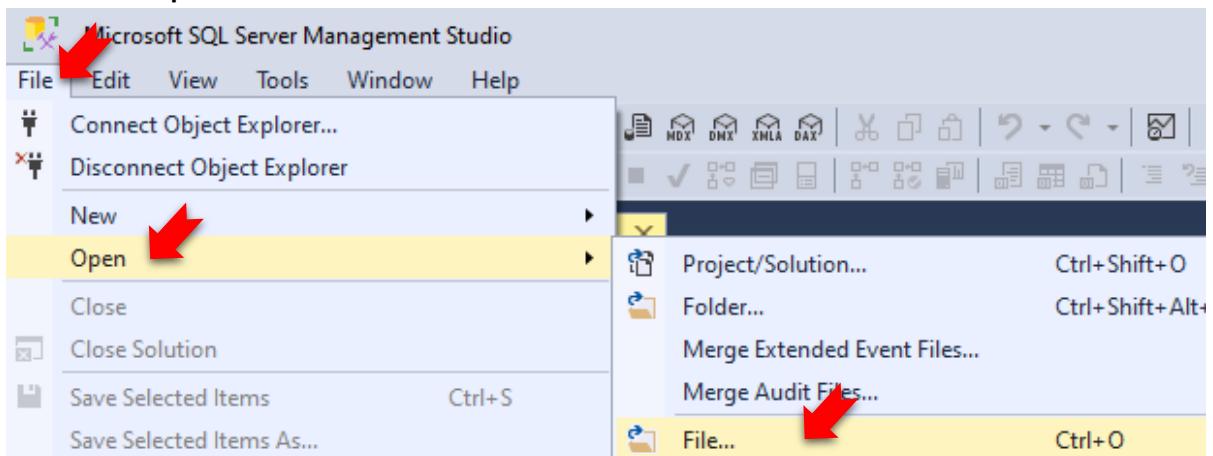


## Prepare supporting tables for the mini project

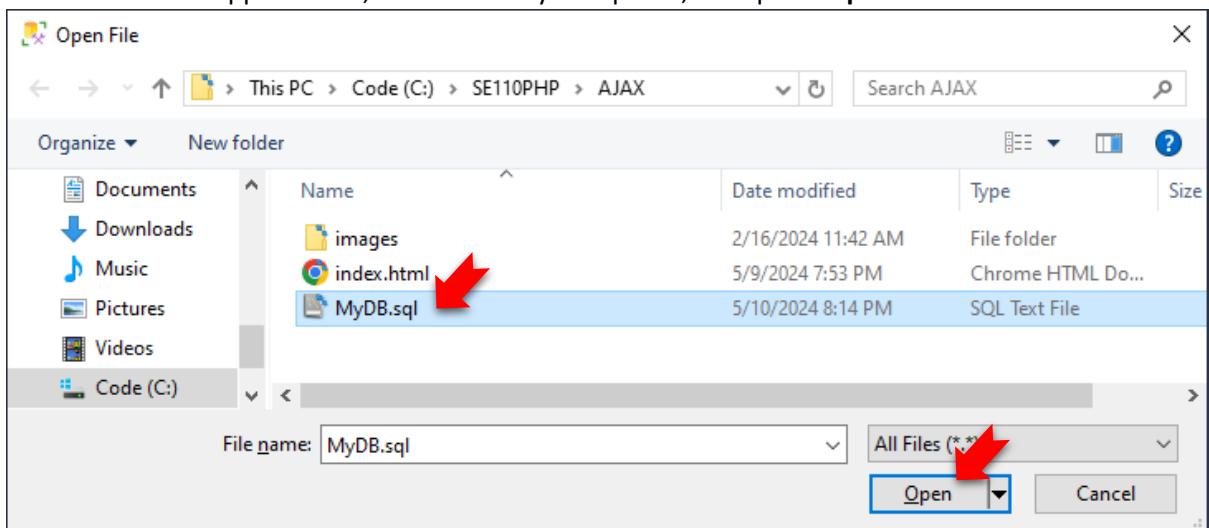
- 1) Start SSMS
- 2) Sign in using **root** account
- 3) If you don't have the database project, create a new one. (In my case, the database is **Ex04**)



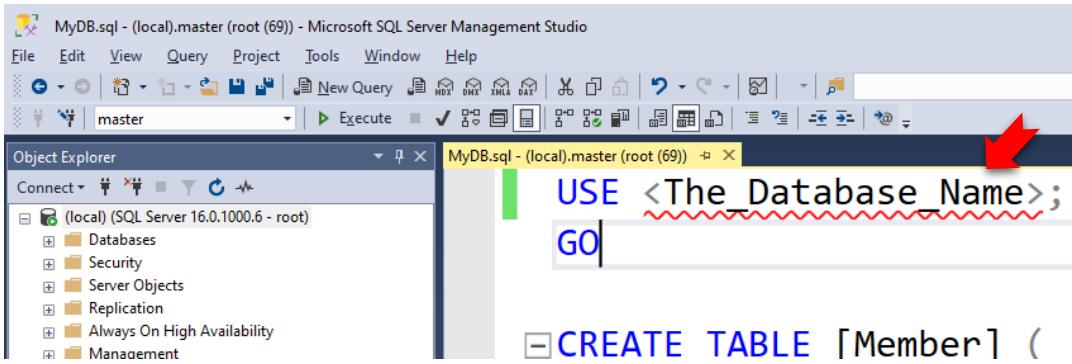
- 4) Select **File->Open->File...**



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



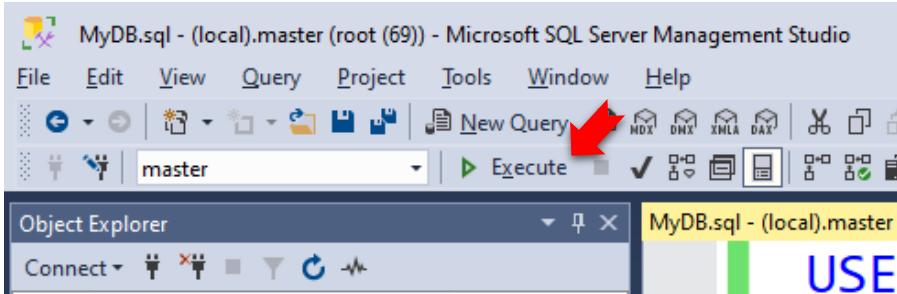
- 6) Replace the database name before running the script



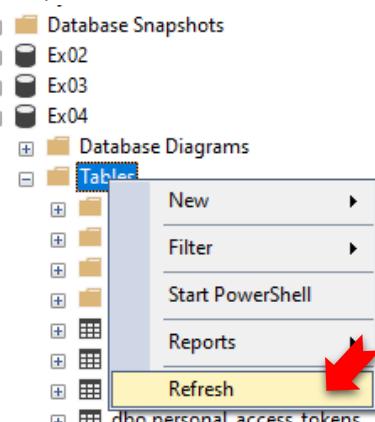
```
USE <The Database Name>;  
GO
```

CREATE TABLE [Member] (

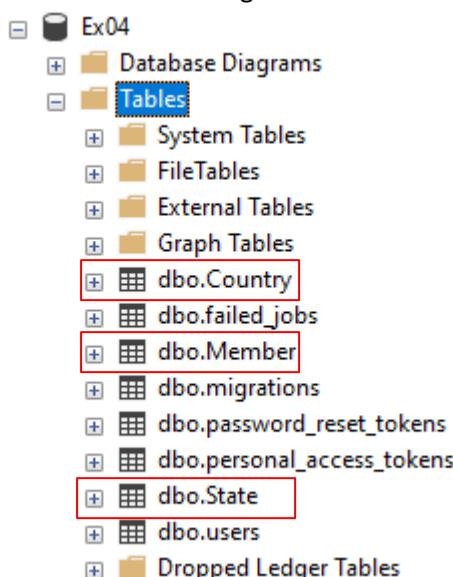
- 7) Make sure that don't select any statement in the script file, then press the **Execute** button



- 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

The screenshot shows the Microsoft SQL Server Management Studio interface. The title bar reads "MyDB.sql - (local).Ex04 (root (69))". Below the title bar, there is a script pane containing the SQL command:

```
USE Ex04;
GO
```

Below the script pane, there are three result sets displayed in tabs:

- Results**: Shows two rows of data from a table with columns Name, Password, and EMail.

	Name	Password	EMail
1	abu	abu123	
2	ali	ali123	
- Messages**: Shows a list of countries with their codes and names.

	Code	Name
1	cn	China
2	jp	Japan
3	kr	Korea
4	my	Malaysia
5	om	Oman
6	sa	Saudi ...
7	sg	Singap...
- Results**: Shows a list of states/provinces with their IDs, country codes, names, and local names.

	ID	Country	Name	LocalName
1	1	my	Selangor	Selangor
2	2	my	Pahang	Pahang
3	3	my	Sabah	Sabah
4	4	my	Sarawak	Sarawak
5	5	my	Perlis	Perlis
6	6	my	Kedah	Kedah
7	7	my	Negeri ...	Negeri S...
8	8	my	Johor	Johor
9	9	my	Penang	Pulau Pin...
10	10	...	...	...

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)

```
C:\ Command Prompt  
Microsoft Windows [Version 10.0.19045.4291]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\HP>cd C:\SE110PHP\Labs
```

A screenshot of a Command Prompt window. It shows the standard Windows welcome message. Below it, the command "cd C:\SE110PHP\Labs" is entered. A red arrow points to the directory path "C:\SE110PHP\Labs".

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

```
C:\ SE110PHP\Labs>composer create-project laravel/laravel:10.0^ Ex04
```

A screenshot of a Command Prompt window. The command "composer create-project laravel/laravel:10.0^ Ex04" is being typed. A red arrow points to the command line.

- 4) Go to the project folder

```
C:\ SE110PHP\Labs>cd Ex04
```

A screenshot of a Command Prompt window. The command "cd Ex04" is being typed. A red arrow points to the directory path "Ex04".

- 5) Run Visual Studio Code from the project folder

```
C:\ SE110PHP\Labs\Ex04>code .
```

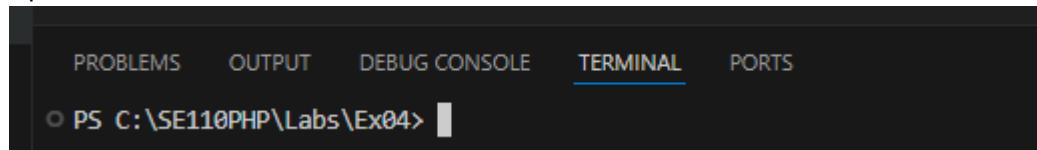
A screenshot of a Command Prompt window. The command "code ." is being typed. A red arrow points to the command line.

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

console.php	10
web.php	11 DB_CONNECTION=sqlsrv
storage	12 DB_HOST=localhost
tests	13 DB_PORT=
vendor	14 DB_DATABASE=Ex04
.editorconfig	15 DB_USERNAME=root
<b>.env</b>	16 DB_PASSWORD="P@\$\$w0rd"
.env.example	
.gitattributes	
.gitignore	

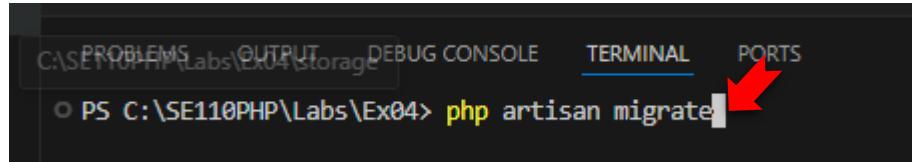
A screenshot of a file explorer. On the left, there is a list of files and folders: "console.php", "web.php", "storage", "tests", "vendor", ".editorconfig", ".env", ".env.example", ".gitattributes", and ".gitignore". The ".env" file is highlighted with a red arrow. On the right, the contents of the ".env" file are shown in a code editor. The file contains several environment variables. Three red arrows point to the right side of the screen, highlighting the variable names "DB\_CONNECTION", "DB\_HOST", "DB\_PORT", "DB\_DATABASE", "DB\_USERNAME", and "DB\_PASSWORD".

- 7) Open the Terminal window



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\SE110PHP\Lab\Ex04>

- 8) Run the migration

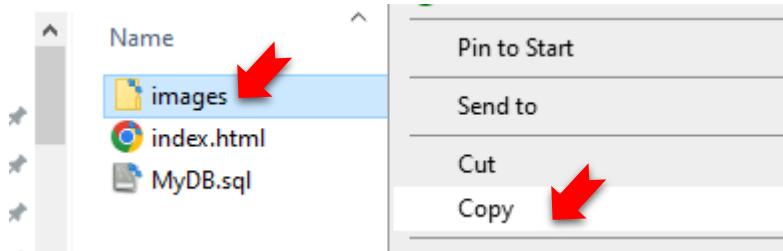


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\SE110PHP\Lab\Ex04> php artisan migrate

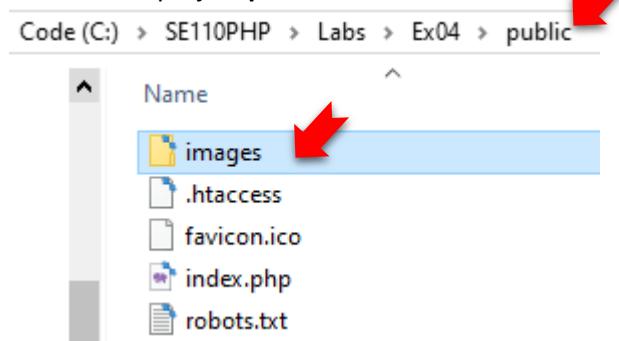
If no error, means the database setting is fine.

### Copy the supporting images

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



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

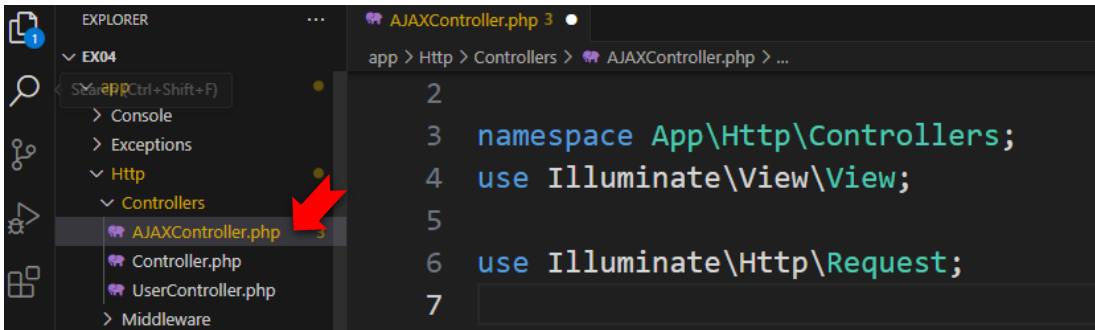


## Create AJAX Controller

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

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

- 2) A new Controller file is created



- 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;
}
```

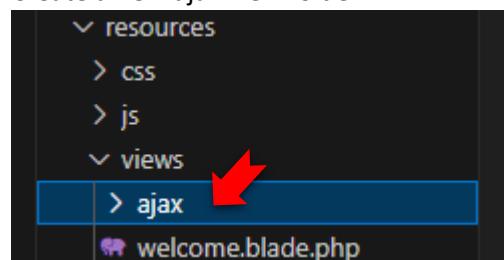
- 8) Add the **GetDays** function the AJAX controller class

```
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++) {
        $result .= "<option>$d</option>";
    }
    return $result."</select>";
}
```

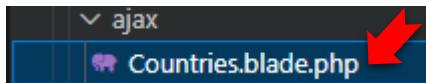
- 9) Save the AJAX control file

### Create the needed view blade files

- 1) Create a new ajax view folder

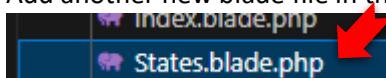


- 2) Add new blade file **Countries.blade.php**



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

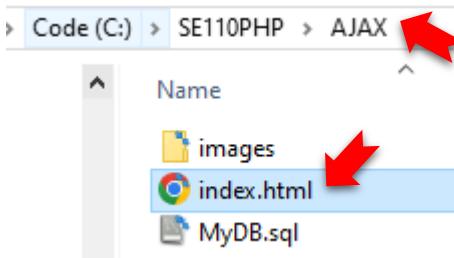
- 3) Add another new blade file in the ajax view folder **States.blade.php**



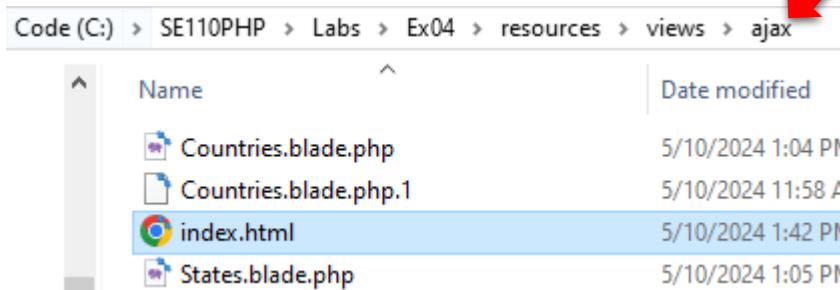
```
<select id='State' name='State'>
    @foreach ($states as $state)
        <option>{{$state}}</option>
    @endforeach
</select>
```

### Prepare the AJAX index blade file

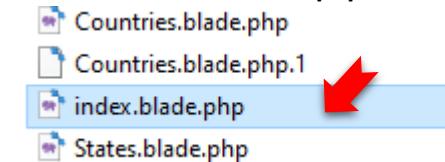
- 1) Copy the index.html file from the unzipped folder



- 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\Jobs\Ex04> php artisan cache:clear
```

```
INFO Application cache cleared successfully.
```

- 2) Another command

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

```
INFO Caching the framework bootstrap files.
```

```
config ..... 46ms DONE  
routes ..... 34ms DONE
```

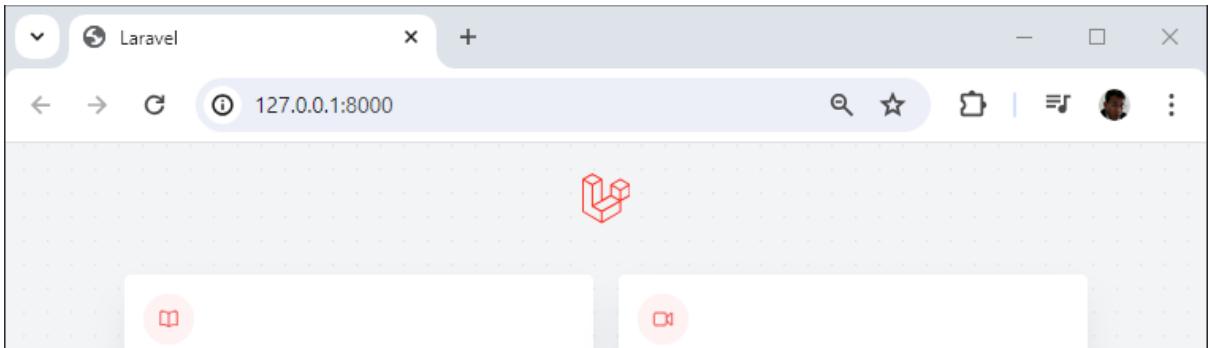
- 3) Start the server, press Ctrl-Click the url

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

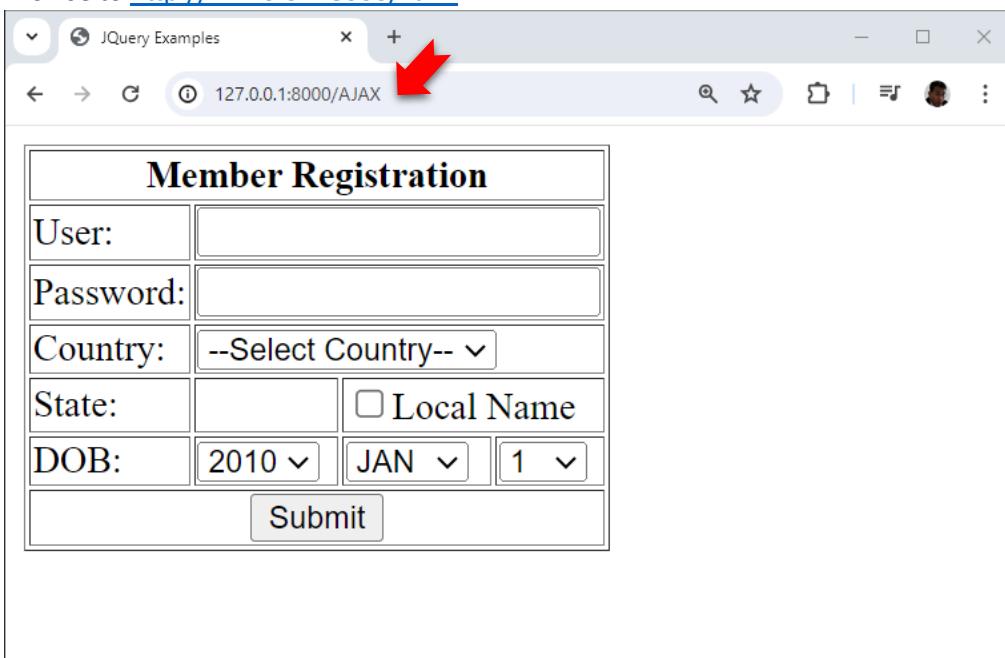
```
INFO Server running on [http://127.0.0.1:8000].
```

```
Press Ctrl+C to stop the server
```

- 4) View the project from browser



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



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

**Member Registration**

User:	Ali 	
Password:	<input type="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:	<input type="password"/>	

**Note:** Meaning that, the user Alibaba is valid

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

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

**Note:** Observe what will happen to the State dropdown list below

Country:	Korea 
State:	Busan 
	<input type="checkbox"/> Local Name

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

State:	부산시 	<input checked="" type="checkbox"/> Local Name 
--------	---	---

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

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