LARAVEL INSTALLATION AND PROJECT CREATION

****

**** THIS FOLDER MUST BE INSIDE XAMPP/HTDOCS *******

****

concept of namespace in normal php file.

1. with same file name and same class inside file, then we have to change the name of a file among those

2. else you can follow these:

a. use namespace Abc; //abc can be replaced with any other name as namespace;

under namespace Abc we are creating a function or everything. Creating namespace same as foldername.

b. using same function and class file copy paste same file into another Pqr folder.

then namespace Pqr; //Pqr is the foldername, and we can't use one.php as require Abc/one.php as because it has to be used with it's namespace.

we can do as: $t=new Abc\one(); // Abc here is the namespace and one() is function

c. we also can use namespace OR import the file with with;

use Abc; //here Abc is namespace.

use Abc\one; //here Abc namespace with class name one.

ALSO CAN BE CREATED OBJECT AS:

$t1=new Abc\one();

$t2=new Pqr\one();

UPLOAD A PROJECT IN GIT WITH COMMAND LINE:

1. MOVE TO YOUR PROJECT FOLDER IN CMD

2. --> git init

3. --> git add .

4. --> git commit -m "first commit" // instead of first commit anything can be written, double quotation is necessary

if(errors after 4th step such as global user & email){

then use step 5 and 6

} else{

skip 5 and 6;

goto 7;

}

5. -->git commit config --global user.email 071.sabin.panthi@gmail.com

6. --> git commit --global user.name 071Sabin

====> 5 and 6 steps are just for the first time, after it's done no need for second time.

7. --> step 4

8. --> git remote add origin <github-repos-link>

9. --> git push <github-repos-link>

10. --> have to login for the first time.

11. --> done!!

BOOTSTRAP

bootstrapmade.com

bootstrap.min.js --. this file has code with eliminated spaces. so min.

1. js has to be loaded just before the body closes.

2. favicon icons??

3. https://getbootstrap.com/docs/5.3/getting-started/introduction/

4. md--> medium devices, col-md-6

5. my ---> margin top and bottom

6. mx --> margin left and right.

7. bootstrap gives browser compatibility

8. <meta> tags are for SEO optimization

Download php and then install composer.

WHEN YOUR PHP FILE DOESN'T RUN, CHECK YOUR PHP VERSION AND CHECK VCC VERSION.

PARTICULAR WINDOWS MAY HAVE PARTICULAR VERSION OF VCC THAT MAY RUN CERTAIN VERSION OF PHP BUT NOT THE HIGHER VERSION.

We may find newer php version not running in our device in future.

SETTING UP FOR LARAVEL INSTALLATION:

1. Download and install **php**, set **environment path** direct to the php folder; path\phpFolder
2. Download and install **composer**, this is what we need to install laravel.

XAMPP/PHP/php-ini *file must be enabled these things: Or* ***php.ini*** *your different, php installed folder.*

//we also can edit the execution times and file uploading sizes etc.

max_execution_time=120

memory_limit=512M

post_max_size=100M

//these must be there(remove semicolon before these, remove comment)

extension=bz2

extension=curl

extension=fileinfo

extension=gettext

extension=mbstring

extension=exif

extension=mysqli

extension=openssl

extension=pdo_mysql

------> *check if these changes are working or not:- http://localhost/dashboard/phpinfo.php*

INSTALLING LARAVEL AND PROJECT CREATION:

1. INSTALL PHP AND COMPOSER FIRST, DEFAULT IT WILL BE INSTALLED IN PHP FOLDER DEFAULT.

2. USING THESE COMMANDS, WE CREATE A PROJECT.

    //installs laravel globally, it's just first time installation in your pc.

    cmd --> **composer global require larvel/installer**

    cmd --> **laravel new <ProjectfolderName>** *//creates project folder.*

3. some errors may arise while executing the first command above. Such as enabling something in php.ini, so go to php installed(not of xampp) because **php.ini** file is edited of the downloaded php, not of Xampp php.

    and then enable those extensions showm in error while installing laravel or creating project.

4. and then go to the project directory just created using command(laravel new <prjname>)

5. after navigating to prj.folder, cmd --> **php artisan serve**

6. SERVER STARTED.....

HOW TO RUN THIS PROJECT?

while using xampp we used localhost:3000..... OR using directly php file as url= **localhost/foldernavigation, but in php:**

1. we use **'artisan'** as to run server in php-laravel.

2.cmd → **php artisan serve(**open the project folder in vsc and then open cmd there.**)**

    SERVER STARTED….

3. After starting the server in cmd, we can open the project folder in vsc and open a new cmd in vsc.

4. open a browser and url= **localhost:8000**

5. laravel simple home will display(or the given path on web.php will display).

6. except public folder everything is secured, all our coding are secured.

FINDING MVC IN YOUR PROJECT FILE;

app/http/controller ===> here will be out controllers, we will inherit from this controller.php.

app/models ===> we will create our models here.

resources/views/anyfile.php ===> this will be views, anyfile inside views folder.

********************these are the three locations where we practice MVC.***

config ========> settings related to DB, file systems, sessions etc.

config/app.php =====> general settings of our project is inside this file.

WRITE CTRL+T IN VSC AND TYPE FILE NAME TO QUICK ACCESS.

WHAT'S INSIDE →CONFIG/APP.PHP?

Inside .env file;

1. env('APP_NAME', 'abc') ====> it means goto .env file and extract APP_NAME if can't extract, use abc.

INSIDE APP.PHP

2. **APP_KEY** ====> random key used for encryption. Everyone has different keys.

3. **APP_DEBUG** = true; //if false it shows 404 or sth else error.

4. **APP_URL** ===> host name.

|***********change timezone in .env file to Asia/Kolkata.***

|***********change APP_NAME=Video_Site in .env file(can't be space in btn video Site).***

---

mysite.com/register.php

mysite.com/login.php

mysite.com/register

MOVE TO ROUTES FOLDER:

THIS HELPS TO NAVIGATE TOWARDS THE FILES. YOU CREATE **ABC.BLADE.PHP** FILE.

THEN ROUTE TO THE SAME **abc.blade.php** file as:

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

****** controller.php controls everything.***

php artisan make:Controller HomeController *// this will create a home controller in the controller.php folder.*

After creating home controller, use/app..... file path.

also create **UserController.php**

create one one functions here in both controller and **usercontroller.php**

It's explained in details below……>>

MANAGING HTML FILES AND IT'S NAVIGATIONS:

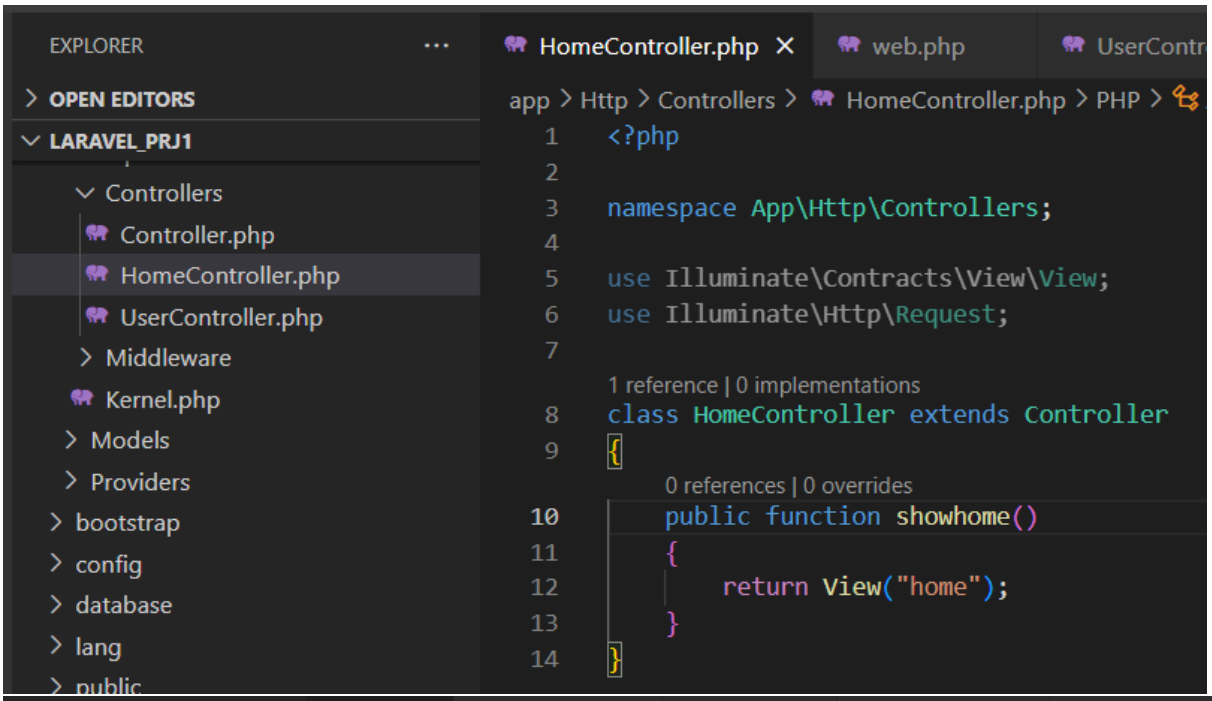
1. Inside **resources/views** in your project folder, you have to create a new file as **filename.blade.php**
2. Write any html code in it, multiple codes can be created.
3. You can create as many files as you want.
4. Suppose you created abc.blade.php file in **resources/views,** THEN ROUTE TO THE SAME **abc.blade.php** file as:
   ```
   Route::get('/', function () {
      return view('abc');
   });
   ```
5. Inside the **routes/web.php**, This code below routes to
   ```
   Route::get('/', function () {
      return view('welcome');
   });
   ```
   Code above explains that it wil be navigated to default **welcome.blade.php** file on **localhost:8000/** loading in url.

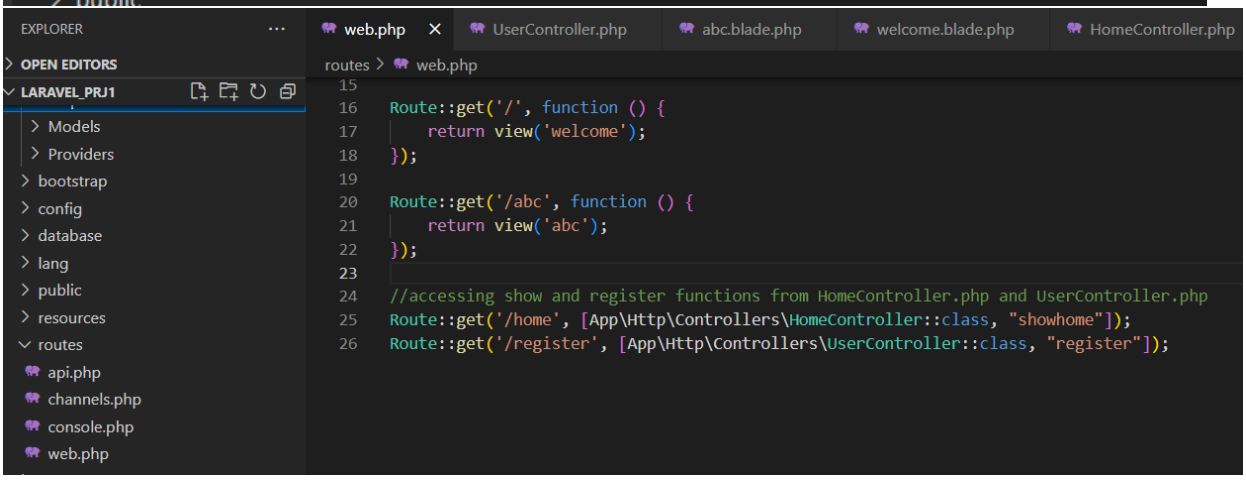
   ```
   Route::get('/register', function () {
      return view('register');
   });
   ```
   To run above code, you must create a **register.blade.php** file inside **resources/views**.
   Like this you can create as many as .blade.php files and link to the **web.php.**

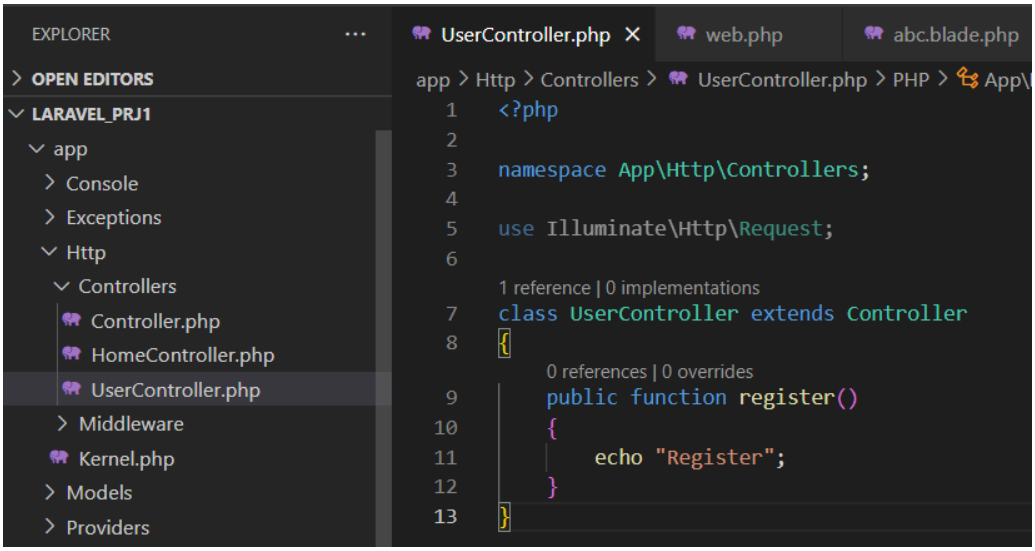
CREATING A NEW CONTROLLERS AND NAVIGATING TO IT'S FUNCTION:

1. Create a new controller from command prompt, as **Controller.php** is the default php file. We are going to create our own controller as:
   php artisan make:Controller HomeController *// this will create a home controller in the controller.php folder. There create your own method and this controller has to be imported using namespace into* ***routes/web.php file****. Homecontroller file extends controller, use it's features with our new HomeControllers. So that no need to edit the default controller. THESE CONTROLLERS ARE CREATED INSIDE* ***app/http/controllers.*** *This path is mentioned in MVC.*

2. Create one more **UserController.php** file and create one more method there.
3. Inside **web.php** file route to the same **HomeController.php** and **UserController.php** file as:
   Route::get('/', [App\Http\Controllers\HomeController::class, "showhome"]);
   Here, **showhome** in above line is the function inside **HomeController.php**.
   Like this many files can be routed. FIG BELOW for showhome function.







This **UserController** file has a **register** function and it is navigated in **web.php** as:
Route::get('/register', [App\Http\Controllers\UserController::class, "register"]);

DAY 3

Vendor folder can be deleted while sharing the project.

- ➔ **Composer.json**=list of libraries that composer has imported
- ➔ **Composer install** can be run in cmd after sharing the file and vendor folder is again created.
- ➔ **Composer update** command is also there, it will look at the list and instead of downloading that library, it will download the latet verison of that library.
- ➔ Creating folder inside controller folder as ➔ cmd—php artisan make:controller user\controllername

CREATED A NEW PROJECT:

Route::get('/user/login', [UserController::class, "login"]);

Click on userController and press ctrl+alt+I to import the class usercontrollre.php automaticaly.

(it's done inside web.php file) pictures are given below:::---

*debugging function dd==> dump and die.

*say to print just x as: {{ x }}

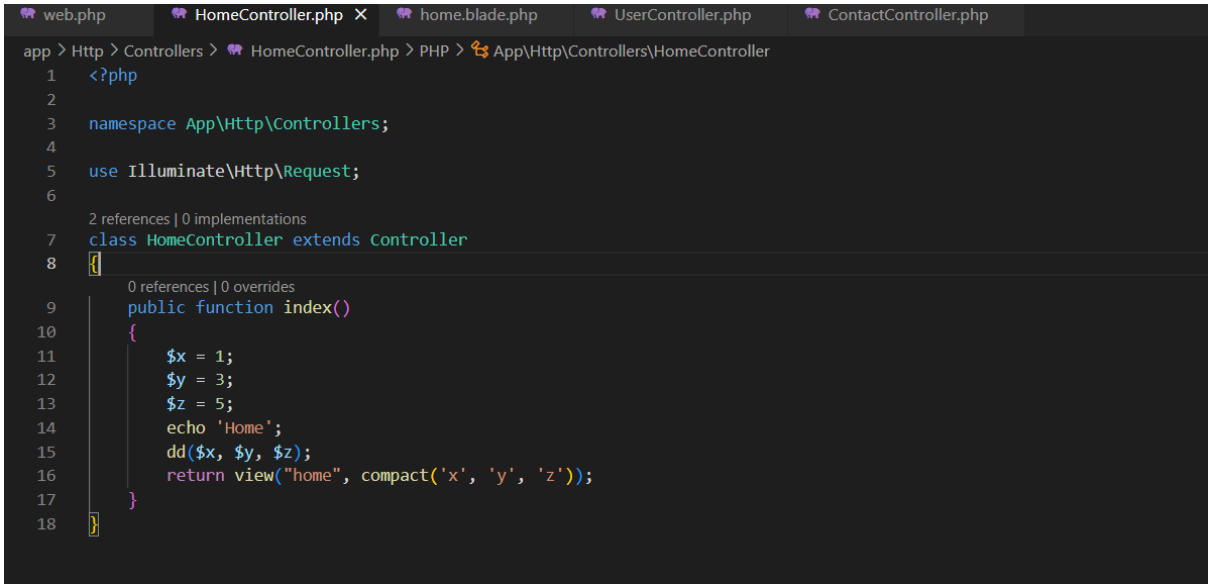
*.blade.php file filters to stop sxecution of js and html files inside a string declaration.

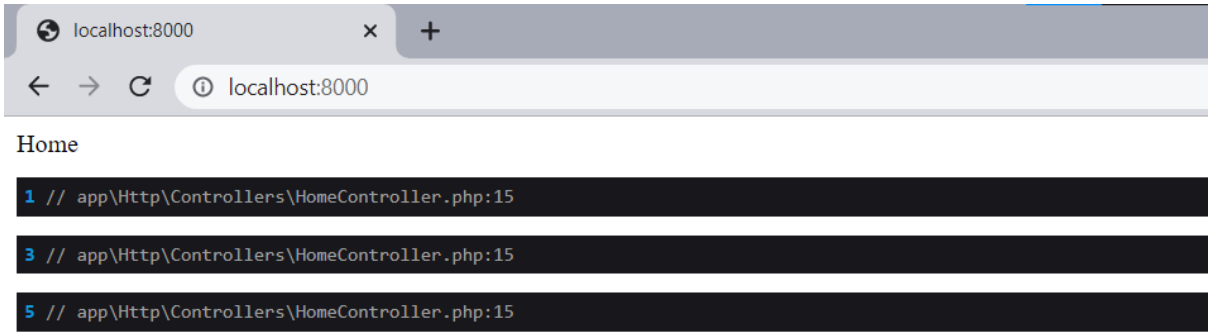
As: $s="sabin <anyJSCode>";

So blade file discards this.

*inside blade file writing if else condition as:

@if($1==1)





YOU CAN SEE EVERYTHING IN LARAVEL WEBSITEWHATEVER SYNTAX ARE GIVEN BELOW

Thse are written inside blade.php file.

@if($x==1)

Hey 1

@elseif(aaa)

Hello aaa

FOR LOOPS:

@for($i=1; $i<10; $i++)

*copy the index.html codes from bootstrap and paste in home.blade.php.

*paste your assets folder from bootstrap_templates. Paste it in public folder where index.php Is there. As runs the index.php.

*laravel suggests absolute paths.

*master page system.

Write this in master.blade.php where you want any other section from home.blade.php.

*@yields('content') // here content is the name of a section

*extends('master') //this extends the master.blade.php file. Both master and home must be in same resources folder.

→ Inside home file, start section as:

@section('content')

    Whatever forms or any div tags html goes here.

@endsection

- ➔ Action can be given to any website name so the request must be from our own site, the request coming from outside should be prohibited.
- ➔ Go to laravel file -> validations to check all the validating formats for a form.
- ➔ Validating form fields: