Contents

[Installation 1](#_Toc45399343)

[Git hub 2](#_Toc45399344)

[Difference between single quote and double quote 2](#_Toc45399345)

[Echo 2](#_Toc45399346)

[Constant 3](#_Toc45399347)

[Loop 3](#_Toc45399348)

[While 3](#_Toc45399349)

[Target 3](#_Toc45399350)

[CRUD Operation in php using mysqli 3](#_Toc45399351)

[I. Create(Insert) 3](#_Toc45399352)

[a. Algorithm 3](#_Toc45399353)

[Session 4](#_Toc45399354)

[Important note: 4](#_Toc45399355)

[Image upload 4](#_Toc45399356)

[File Delete or Unlink 5](#_Toc45399357)

[Edit Image 5](#_Toc45399358)

[Explode 5](#_Toc45399359)

[In\_Array 6](#_Toc45399360)

[Laravel 6](#_Toc45399361)

[Folder Structure 8](#_Toc45399362)

[.env 8](#_Toc45399363)

[Composer.json 8](#_Toc45399364)

[Package.json 8](#_Toc45399365)

[Process 8](#_Toc45399366)

[Login and register 9](#_Toc45399367)

[Table in Laravel: 10](#_Toc45399368)

[Create a table in laravel 11](#_Toc45399369)

# Installation

Long ago we used to work in Xampp and Wamp for the local server now we use Laragon

It’s easy to use than other software.

1. Easy to create virtual domain
2. Gives ssl certificate
3. Easy to open project from www
4. Easy for laravel project don’t need to write cmd: php artisan serve

# Git hub

1. Create a new Repository
2. git init
3. git status
   * 1. red: commit is still not done
     2. green: commit has been done
4. git remote add origin git@github.com:krevilraj/Simple-CRUD.git
5. git add .( it will change into green)
6. git commit -m “ message” after this all files will be green now it is ready for push
7. git push origin master

# Difference between single quote and double quote

<?php

$x = "Hello world!";

$y = 'Hello world!';

echo $x;

echo "<br>";

echo $y;

?>

Single quote and double quote both used for string. But Double quote recognize variable whereas single quote doesn’t.

# Echo

It only recognizes variable, single quote, double quote, concatenation, Athematic operator,

Only var\_dump and print\_r() recognize array

<?php

$x = "Hello world!";

$y = 'Hello world!';

echo $x;

echo "<br>";

echo $y;

?>

# Constant

<?php

define("GREETING", "Welcome to W3Schools.com!");

echo GREETING;

?>

# Loop

## While

First check condition then if condition meet or True then code inside the bracket code will be execute otherwise exit the loop.

while (*condition is true*) {

*code to be executed*;

}

# Target

1. PHP basic
2. PHP database
3. Database Query
4. Laravel
5. Jquery
6. Vue

# CRUD Operation in php using mysqli

## Create(Insert)

### Algorithm

* Create Form
* Form method and action is necessary. “method”: POST, GET, PUT and DELETE. “action”: Where to go after submit is click.
* Check Post data
  + 1. echo $\_POST[‘name’]
    2. var\_dump($\_POST);
    3. exit();
* Insert into database

$name = $\_POST['name'];

    $address = $\_POST['address'];

mysqli\_query( $db, "INSERT INTO info (name, address) VALUES ('$name', '$address')" );

$\_SESSION['message'] = 'Address saved';

    header('Refresh: 1; URL=add.php');

# Session

message should be pass one page to another page for this purpose we use session

example:

if we need user information in all page then we use session then print session

* Initialize session or store session

$\_SESSION['user\_login'] = true;

* Print Session

echo $\_SESSION['message'];

* + Delete Session or Unset Session

unset($\_SESSION['message']);

# Important note:

* Do not forget quote in query
* Var\_dump exit

# Image upload

Note:

* 1. In form

enctype="multipart/form-data"

* 1. Input type must be file

<input type='file' name='image' />

* 1. Jastai timlai validation pdf file matra upload kasari

if(isset($\_FILES['image'])){ // input type file is taken as $\_Files

        $errors= array();

        $file\_name = $\_FILES['image']['name'];

        $file\_size =$\_FILES['image']['size'];

        $file\_tmp =$\_FILES['image']['tmp\_name'];

        $file\_type=$\_FILES['image']['type'];

        $extensions= array("jpeg","jpg","png","pdf");

        $tmp = explode('.', $file\_name);

        $file\_ext = end($tmp);

        if(in\_array($file\_ext,$extensions)=== false){

           $errors[]="extension not allowed, please choose a JPEG or PNG file.";

        }

        if($file\_size > 2097152){

           $errors[]='File size must be excately 2 MB';

        }

        if(empty($errors)==true){

           move\_uploaded\_file($file\_tmp,"images/".$file\_name);

           echo "Success".$file\_name;

        }else{

           print\_r($errors);

        }

     }

# File Delete or Unlink

$path = $\_SERVER['DOCUMENT\_ROOT'].'\/images\/'.$file\_name;

unlink( $path );

# Edit Image

If(new iimage is uploaded) =>old image should be deleted and new image should be uploaded

Else no new image => nothing change

# Explode

<?php

$str = "Hello world. It's a beautiful day.";

print\_r (explode(" ",$str));

?>

# In\_Array

# Laravel

In laravel almost everthing is done by package(like in wordpress plugin). All package resides in vendor folder.

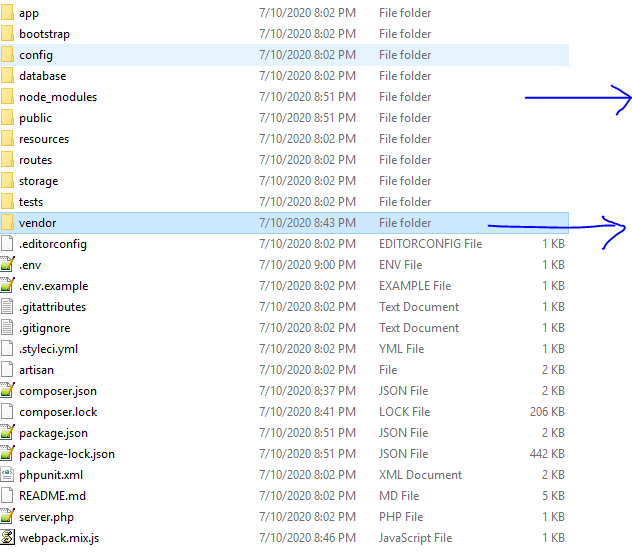
Requirement

If we install laragon it will fulfill all the requirement.

If you xampp and wamp

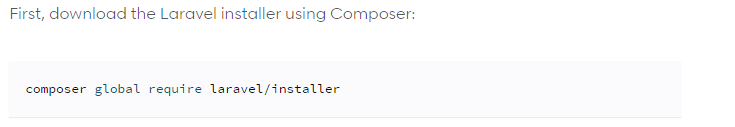
* PHP >= 7.2.5
* BCMath PHP Extension
* Ctype PHP Extension
* Fileinfo PHP extension
* JSON PHP Extension
* Mbstring PHP Extension
* OpenSSL PHP Extension
* PDO PHP Extension
* Tokenizer PHP Extension
* XML PHP Extension

We need composer to install laravel



Npm install

Composer install



Composer will download template of laravel

Now we can write

laravel new blog

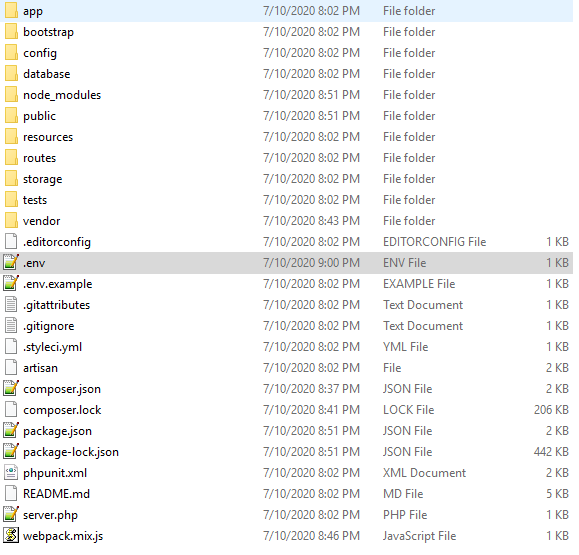
it will create new laravel project with folder name blog

Creating a laravel project

composer create-project --prefer-dist laravel/laravel blog

where blog is project name

# Folder Structure



## .env

Database connection

## Composer.json

All package is declared here.

## Package.json

All node dependencies are declared here.

# Route

Api: Api related route are defined here

Web: Web related route are defined here

# Resource:

All the frontend html or php page stays in views folder.

# Process

* Cmd => composer create-project --prefer-dist laravel/laravel blog
* .env file ma database connection
* Create database
* To run the project we write a command

php artisan serve

* composer require laravel/ui
* php artisan ui vue --auth
* npm install && npm run dev (design problem will be solve)

# Login and register

We don’t have auth folder in

learninglaravel\app\Http\Controllers

learninglaravel\resources\views

in web.php

only

Route::get('/', function () {

return view('welcome');

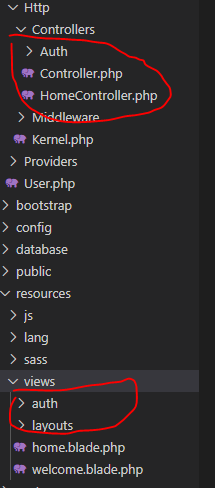
});

Laravel gives us login and register by default but first we have write

composer require laravel/ui

php artisan ui vue --auth

Changes appear in



and also in route/web.php

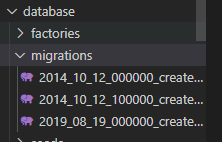
Auth::routes();

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

# Table in Laravel:

Database table is not manually created in the database.

All table is written in migration folder.



Cmd:

php artisan migrate

this command will run all the migration files

it creates 3 table in database with migration table as well.

## Create a table in laravel

php artisan make:migration create\_addresses\_table

It creates migration file in migration table.

public function up()

    {

        Schema::create('addresses', function (Blueprint $table) {

            $table->id();

            $table->string('name');

            $table->string('address');

            $table->string('image');

            $table->timestamps();

        });

    }

# Edit table in laravel

If we want to edit table we can’t update migration file and run php artisan migrate

If we want to add phone column in addresses table

Old ways:

Update the migration file we add phone column

Then we run

php artisan migrate

we get nothing to migrate (Error)

IN Laravel :

There are two ways:

1. First way (Disadvantage it delete all the records)

* Delete or drop the existing table (addresses table)
* Delete the record of migration file in migration table

Export to save all data -> use first way -> table with extra column->import all

1. Second way

* First we create migration file (php artisan make:migration add\_age\_in\_address\_table)
* We edit the file
* Up (add , remove, alter)
* Down(remove)

 public function up()

    {

        Schema::table('addresses', function (Blueprint $table) {

            $table->string('age')->after('address');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::table('addresses', function (Blueprint $table) {

            $table->dropColumn('age');

        });

    }

Condition

* + - 1. Project is in development (Still not host and you are the only one developer)

First way or second way

After someday mobile number want to be added

If there are other developer than you cannot do First way.

Simply pull from git and run php artisan migrate

* + - 1. Project running

Second way is better option

# Error:

Design and database connection is still remaining.(npm install && npm run dev will solve vue design problem)

SqlState: php artisan migrate

**SQLSTATE[HY000]: General error: 1364 Field 'phone' doesn't have a default value**

Name of this error is default value:

Reason of this error

There is no default value for phone

# Login and Register

Laravel gives us login and register by default but first we have write

composer require laravel/ui

php artisan ui vue --auth

# Laravel Actual process:

First browser go to route-> web.php

Then it search for url segment in web.php

Route::get('/', function () {

    return view('welcome');

});

Auth::routes();

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

Route::get('/raj', function () {

    echo "rajkumar";

});

If it finds the url segment then it redirect to controller or return view or just echo.

# Page View process:

1. Write in the route (web.php) then return the view

Route::get('/add-address', function () {

    return view('add-address');

});

1. Write the controller name to redirect in web.php

Route::get('/all-address', 'HomeController@all'); //

And in HomeController.php

public function all(){

        return view('all-address');

    }

1. Create all-address.blade.php file in view folder