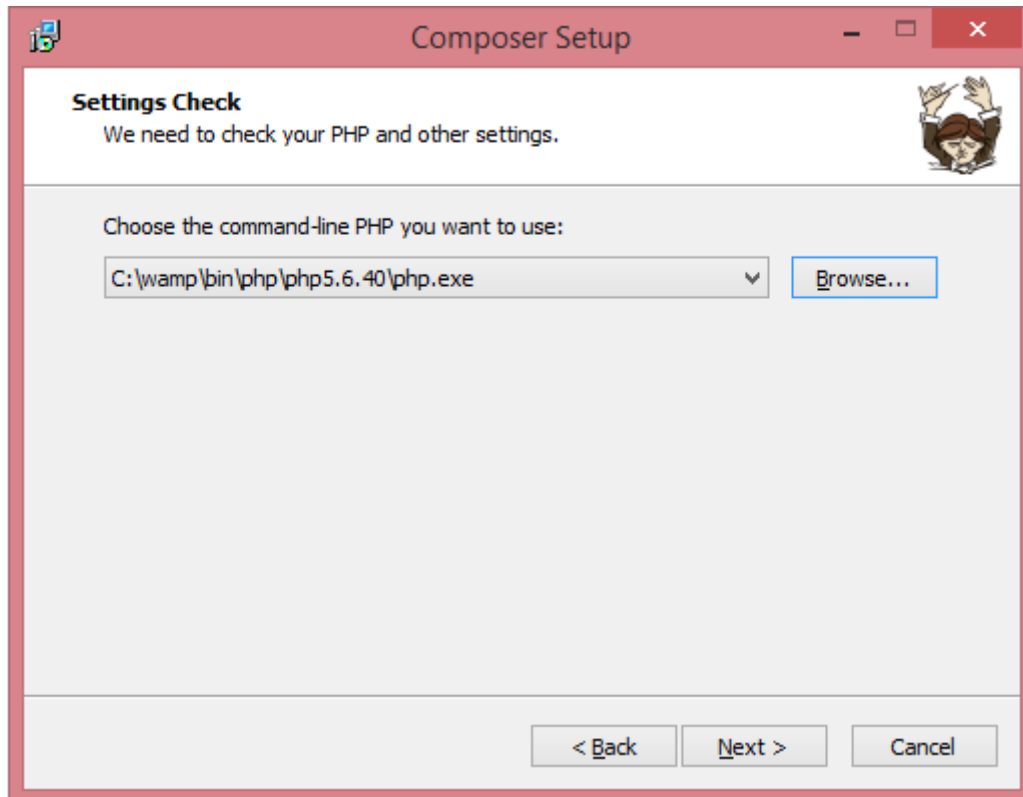


1.Download composer and wampp/xampp if not installed

composer : <https://getcomposer.org/download/>

xampp : <https://www.apachefriends.org/download.html>

wampp : <https://www.wampserver.com/en/>



2.Provide the path to php.exe in xampp/wampp for composer

3.Download git and vscode

<https://code.visualstudio.com/download>

<https://git-scm.com/downloads>

4.Use composer create-project laravel/laravel firstproject to create a project named firstproject.

Create the project in htdocs folder for xampp and www folder for wampp

5.Install auth package

- composer require laravel/ui
- php artisan ui bootstrap
- php artisan ui bootstrap --auth

6.Download node.js to run the auth

<https://nodejs.org/en/download/>

7.install and run npm

- npm install
- npm run dev

8.Basics

All Laravel routes are defined inside the route files located in the routes directory. When we create a project, then a route directory is created inside the project.

The route/web.php directory contains the definition of route files for your web interface.

```
<?php
Route::get('/', function ()
{
    return view ('welcome');
});
```

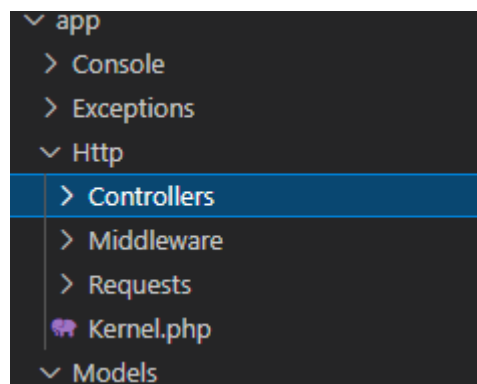
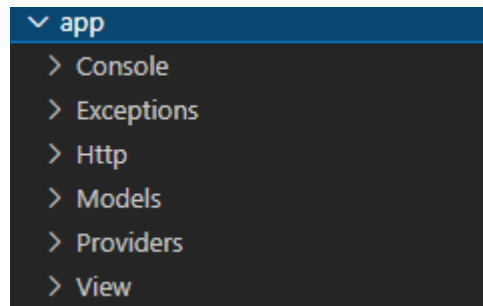
In the above case, Route is the class which defines the static method get(). The get() method contains the parameters '/' and function() closure. The '/' defines the root directory and function() defines the functionality of the get() method.

```
Route::get('user/{id?}', function ($id=null) {
    return "id is : ". $id;
}->where('id','[0-9]+');
```

- Above only allows integer value

```
Route::get('user/{name?}', function ($name=null) {
    return $name;
})->where('name','[a-zA-Z]+');
```

- Above only allows alphabetic value
- From the web.php in route file the function is called from the app/http/controller/controllername.



- Whatever we have to return or store we do that in the controller. The controller must be connected with its adjacent model which helps to connect with the database.

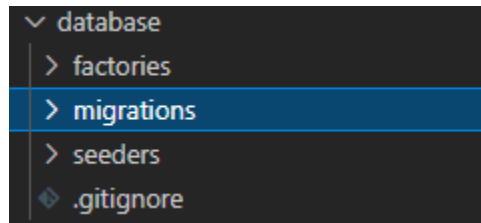
```
use App\Models\User;
```

For ex. We must include the user model while working in usercontroller

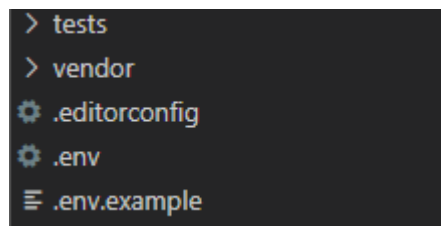
- Similarly you have to include path to the controller in web.php

```
use App\Http\Controllers\UserController;
```

- The tables are in database/migration folder.



- We have to make changes in .env file to make connection with the database.



- In env file, find this and make changes accordingly

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=port number(usually it is 3306)
DB_DATABASE=Database name
DB_USERNAME=root
DB_PASSWORD=
```

- You must put all your assets like css,js,pictures in the public folder.
- The index.blade.php in the public folder is the first page to open.
- You have to include .blade in your filename ex.name.blade.php. Blade templates makes so that we don't have to write php code in <?php ?>.
- If you have already installed the admin package then the home.blade.php will open after the login process.
- The welcome.blade.php is the homepage that will open if we just open the

webpage.

- If you want any data to be shown or to use a function put it in double inverted brackets `{{}}`
- If you want to provide path to css or js file just use `asset()` function and provide the path in public folder. The `asset()` function will take you to the public folder.

```
<link rel="stylesheet" href="{{asset('dist/css/adminlte.min.css')}}">
```

- The `route()` function will take you to the defined route that provided the it is already defined in `web.php`. It is usually used to go to particular function in controller.

```
action="{{route('files.store')}}"
```

- You must never forget to include `enctype="multipart/form-data"` while making form
- The `@csrf` token must be included in form to differentiate different data.
- The `redirect()` function will redirect you to provided web link. It is usually used to go to particular page in `resource/views/` folder.

```
return view('foldername-in-view.pagename');
```

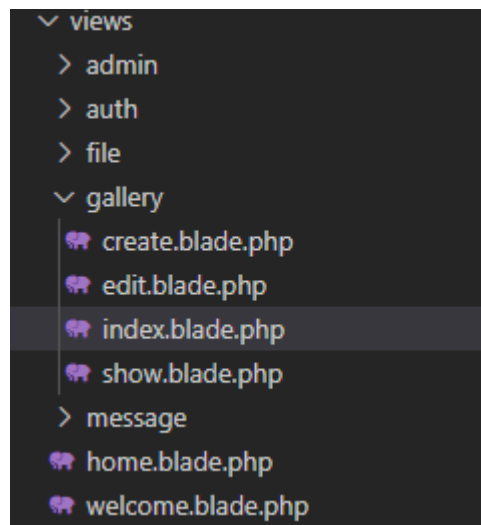
- Or you can even use route with redirect

```
return redirect()->route('name-in-route/web.function-name');
```

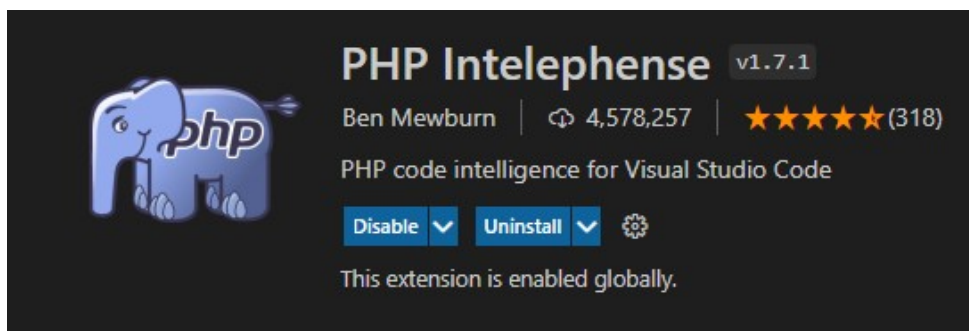
- you must put `→middleware('auth')` after calling a class in `web.php`

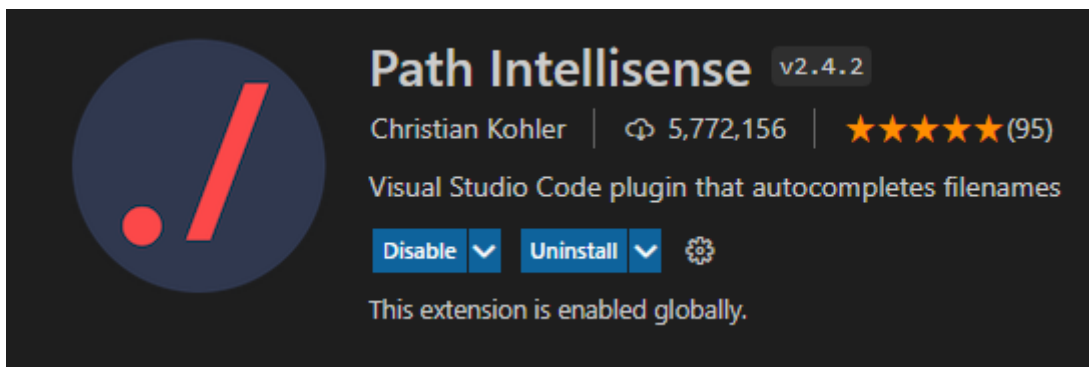
```
route::resource('files',FilesController::class)->middleware('auth');
```

- make individualized folder for all elements in table in `resource/views` folder



6.Packages for vscode





7.Creating resource file

php artisan make:model message -mcr

- By using this code it will make a model,controller and a migration page

```
route::resource(['name-to-be-used-in-route',controllername::class]);
```

- code for web.php
- You must also include the path to the controller

```
use App\Http\Controllers\controllername;
```

- The route paths

HTTP Verb	Path (URL)	Action (Method)	Route Name
GET	/sharks	index	sharks.index
GET	/sharks/create	create	sharks.create
POST	/sharks	store	sharks.store
GET	/sharks/{id}	show	sharks.show
GET	/sharks/{id}/edit	edit	sharks.edit
PUT/PATCH	/sharks/{id}	update	sharks.update
DELETE	/sharks/{id}	destroy	sharks.destroy

- index is the function which is called first
- create is the function to redirect to the page where we fill in the data
- store is function where the data entered in create page is sent to be stored in the database

- show is function to redirect to the page used to show individual data
- edit is the function to redirect to page to enter new updated data
- update function is where the data is updated in the database
- destroy function is where the data is deleted from database

➤ We have to create individual form to delete option to use route() function

```
<form action="{route('Gallery.destroy', $gallery->id) }}" method="POST">
    @csrf
    @method('DELETE')
    <button class="btn btn-danger btn-sm" type="submit">Delete</button>
</form>
```

8.index()

In index we return the page to the manage page and provide value to display in that page.

We send back the value either using with() or compact()

```
return view('student')->with('id',$id);
```

```
$galleries=Gallery::latest()->get();
return view('gallery.index',compact('galleries'));
```

- To paginate the results

```
$galleries=Gallery::latest()->paginate(3);
return view('gallery.index',compact('galleries'));
```

latest() extracts the latest data and paginate() is used to show the no of data inside the paginate() function

- Put

in boot() in app/providers/appserviceprovider.php

and `use Illuminate\Pagination\Paginator;`

on the top of the page

- Put

```
<tfoot>
    <td>{{ $galleries->onEachSide(5)->links() }}</td>
</tfoot>
```

in the index page. `onEachSide()` is used to show the no of pages on each side putting the current page in middle. `links()` automatically create link to the pages.

9.search

- Put it in index

```
<form action="{{ route('search') }}" method="GET">
    <input type="text" class="col-lg-10" name="search" required/>
    <button class="btn btn-primary btn-sm mb-1" type="submit">Search</button>
</form>
```

- In `web.php`

```
Route::get('/search/', [App\Http\Controllers\GalleryController::class, 'search'])->name('search');
```

- In the controller

```
public function search(Request $request){
    // Get the search value from the request
    $search = $request->input('search');

    // Search in the title and body columns from the posts table
    $galleries = Gallery::query()
        ->where('name', 'LIKE', "%{$search}%")
        ->orWhere('pic', 'LIKE', "%{$search}%")
        ->paginate(3);

    // Return the search view with the results compacted
    return view('gallery.index', compact('galleries'));
}
```

instead of name and pic in `where()` put the fields you want to make the searchable.

10.create()

For create just redirect it to the create page in view folder

```
return view('file.create');
```

11.store(argument)

- Before storing,you can also validate the data
- For storing file

```
$filename = str_replace(' ','',request('name'));  
$ext = $request->filelink->extension();  
$finalname = $filename.'_'.time().'.'.$request->filelink->extension();  
$request->filelink->move(public_path('uploads/files/'),$finalname);  
  
$files = new Files();  
$files->name = $request['name'];  
$files->filelink = $finalname;  
$files->ext = $ext;  
$files->save();  
  
return redirect()->route('files.index');
```

First of all we remove the space using str_replace() then extract the extension , add time to make the data different even if they have same value.Move the file to uploads/files folder in public folder.Lastly we save the data in the database.

- For storing other data

```
$galleries=new Gallery();  
$galleries->name = $request['name'];  
$galleries->details=$request['details'];  
$galleries->pic = $request['pic'];  
$galleries->save();  
  
return redirect()->route('Gallery.index');
```

12.edit()

Same as create, we send the page to edit page after finding the

correct data by compairing id.

```
$galleries=Gallery::find($id);  
return view('gallery.edit',compact('galleries'));
```

13.update()

Find the data by compairing id and update the data in database

```
$request->validate([  
    'name' => 'required|max:255',  
    'details' => 'required|max:255',  
    'filelink' => 'required|mimes:png,jpg,jpeg'  
]);  
  
$galleries=Gallery::find($id);  
$galleries->name = $request['title'];  
$galleries->img = $request['img'];  
$galleries->details=$request['details'];  
$galleries->update();  
  
return redirect()->route('Gallery.index');
```

14.delete()

Find the id and delete from the database

```
$Galleries = Gallery::where('id', $id)->first();  
$Galleries->delete();  
return redirect()->route('Gallery.index');
```

15.models

```
use HasFactory;  
protected $fillable=[  
    'name','filelink','ext','status'  
];
```

In the place of name,filelink etc put the titles you used in the database.

16.migrations

```
public function up()
{
    Schema::create('files', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('filelink');
        $table->string('ext');
        $table->integer('status')->default(1);
        $table->timestamps();
    });
}
```

In migration we mainly create the table for database and migrate it using:

```
php artisan migrate
```

for individual table use:

```
php artisan migrate --path The path to the table by right clicking on table and copying path
```

While copying path only copy from database\

```
ex database\migrations\2021_10_30_022423_create_galleries_table.php
```

17. layouts

The layout folder is already created with the auth package.

You make a layout and use the same layout for all pages using section and extends function

```
@extends('admin.layouts.layout')
@section('content')
```

We use @yield(name) in the layout page which extracts the data from the section(name) from the page where we extend the layout using @extends(path to layout)

```
@yield('content')
```

You have to end the section after filing data using @endsection

```
@endsection
```

18.edit page

```
@csrf  
@method('PUT')
```

You use put method in the edit page to insert the data and show the data in value field. Never for to put csrf in the form.

```
value="{{ $galleries->name }}"
```

19.Index page

Use foreach(var-returned-from-controller as new-var-name)

You should put the value inside {{ }} and instead of name, filelink use the name you used in table.

```
<table class="table table-striped table-bordered sourced-data">  
  <thead>  
    <tr>  
      <th>Action</th>  
      <th>Name</th>  
      <th>File_Link</th>  
      <th>Extension</th>  
      <th>Status</th>  
    </tr>  
  </thead>  
  @if($files)  
  @foreach ($files as $file)  
    <tbody>  
      <tr>  
        <td>  
          <form action="{{ route('files.destroy', $file->id) }}" method="POST">  
            @csrf  
            @method('DELETE')  
            <button class="btn btn-danger btn-sm" type="submit">Delete</button>  
          </form>  
        </td>  
        <td>{{ $file->name }}</td>  
        <td>{{ $file->filelink }}</td>  
        <td>{{ $file->ext }}</td>  
        <td>{{ $file->status }}</td>  
      </tr>  
    </tbody>  
  @endforeach  
@endif  
</table>
```

Activate
Go to Settings