

composer-> to install php dependencies other, you'll have to install all the dependencies manually.

// create a laravel project by running the following command;

composer create-project laravel/laravel projectName

// run the project by running the following command;

Php artisan serve

//create a controller using the following command to display the value of a variable;

Php artisan make:controller ProductsController

// create a function to display the value of a variable;

```
public function display(){
    return view('new', ['testVar'=>'Tooba!']);
}

public function display(){
    return View('index', ['name'=>'Zaid']);
}

public function display(){
    $testVar = "Toobaaa!";
    return view('new', ['testVar'=>$testVar]);
}

public function display(){
    $testVar = "ToobaaaKhawar!";
    return view('new', compact('testVar'));
}
```

// give route for the view 'display' that shows the value of the variable

```
Route::get('/showVar', [ProductsController::class, 'display']);
```

// embedding a template
// copy/paste header/footer inside the

//create a view by right clicking on the views folder which is inside the resources folder.

_layout.blade.php

```
@yield('something')
```

Write the following commands on the pages that extend the _layout.blade.php

```
@extends('_layout')
@section('something')

// Paste the copied content from the template here

@endsection
```

// give routes of all the pages created.

```
Route::get('/', function () {
    return view('welcome');
});
Route::get('/dashboard', function () {
    return view('dashboard');
});
Route::get('/about_temp', function () {
    return view('about');
});
Route::get('/gallery', function () {
    return view('gallery');
});
Route::get('/gallery', function () {
    return view('gallery');
});
Route::get('/people', function () {
    return view('people');
});
```

// now, provide links into the navbar , inside the -layout.blade.php file.

```
<li><a href="{{url('dashboard')}}" class="">Home</a></li>
  <li><a href="{{url('about_temp')}}">About</a></li>
  <li class="dropdown"><a href="#"><span>Gallery</span> <i class="bi bi-
chevron-down dropdown-indicator"></i></a>
    <ul>
      <li><a href="{{url('gallery')}}">Nature</a></li>
      <li><a href="{{url('people')}}">People</a></li>
```

➔ The template is embedded!

// create database using phpMyAdmin / MySQL

RENAME THE DATABASE INSIDE THE .ENV FILE

- ➔ .ENV stands for (environment specific variables for the app)
- ➔ Php artisan migrate
- ➔ For user authentication (login/signup) run the following commands

```
➔ <!-- composer require laravel/ui
➔ php artisan ui bootstrap --auth
➔ npm install (agar nodejs installed nai hoga to ye run nai hoga!)
➔ npm run dev
➔ npm run development OR npm run build -->
➔ php artisan serve
```

//create migrations

php artisan make:migration create_students_table

// create model

php artisan make:model students

// create controller

php artisan make:controller StudentsController

//make migration along-with model class

php artisan make:model students -m

// make migration along-with model class and controller

php artisan make:model students -m -c

// create a migration for students including controller and model class

Inside the migration give columns inside the UP function;

```
public function up()
{
    Schema::create('students', function (Blueprint $table) {
        $table->id('pro_id');
        $table->string('pro_name');
        $table->string('pro_desc');
        $table->integer('pro_price');
        $table->timestamps();
    });
}
```

// run command php artisan migrate

//Create a view by right clicking on the view folder and create a form

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>CREATE RECORD</title>
</head>
<body>
  <form action="/save" method="post">
    @csrf
    <!-- (Cross-site Request Forgery) -->
    <input type="text" name="pro_name" placeholder="Product Name">
    <input type="text" name="pro_desc" placeholder="Product Description">
    <input type="number" name="pro_price" placeholder="Product Price">
    <input type="submit" name="ADD PRODUCT">
  </form>
</body>
</html>
```

// Controller

```
public function save(Request $req){

$pro_name = $req -> input('pro_name');
$pro_desc = $req -> input('pro_desc');
$pro_price = $req -> input('pro_price');

Product :: create(['pro_name'=> $pro_name, 'pro_desc'=>$pro_desc, 'pro_price'=> $
pro_price]);
return redirect()->action([ProductsController::class,'getdata']);
// return "Product Added!";
}
```

//route

```
Route::post('/save',[ProductsController::class,'save']);
```

```
// display records
```

```
//Controller
```

```
public function getdata(){

$product = Product::all();
    <!-- return view('display',compact('product')); -->
return view('display',['product'=>$product]);

}
```

```
//view
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>DISPLAY RECORDS</title>
</head>
<body>
    <table>

        <tr>
            <th>Product Id:</th>
            <th>Product Name:</th>
            <th>Product Description:</th>
            <th>Product Price:</th>
            <th>Actions:</th>
        </tr>
        @foreach($product as $pro)
            <tr>
                <td> {{ $pro->pro_id }} </td>
                <td> {{ $pro->pro_name }} </td>
                <td> {{ $pro->pro_desc }} </td>
                <td> {{ $pro->pro_price }} </td>
                <td> <a href="Edit/pro_id={{ $pro->pro_id }}">Edit</a> </td>
                <td> <a href="Delete/pro_id={{ $pro->pro_id }}">Delete</a> </td>
            </tr>
        @endforeach
    </table>
</body>
</html>
```

```
// route
```

```
Route::get('/display',[ProductsController::class,'getdata']);
```

```
//edit
//Controller
```

```
public function fetch_id($pro_id){
$editData = Product::where('pro_id',$pro_id)->get();
return view('Edit',compact('editData'));
}

public function update(Request $req){

$id = $req->input('pro_id');
$name = $req->input('pro_name');
$desc = $req->input('pro_desc');
$price = $req->input('pro_price');

Product::where('pro_id',$id)-
>update(['pro_name'=>$name, 'pro_desc'=>$desc, 'pro_price'=>$price]);
return redirect()->action([ProductsController::class,'getdata']);

}
```

```
//view
```

```
<body>
@foreach($editData as $data)
    <form action="/update" method="post">
        @csrf
    <input type="hidden" name="pro_id" value="{{ $data->pro_id }}">
    <input type="text" name="pro_name" value="{{ $data->pro_name }}" >
    <input type="text" name="pro_desc" value="{{ $data->pro_desc }}" >
    <input type="number" name="pro_price" value="{{ $data->pro_price }}" >
    <input type="submit" name="SAVE CHANGES">

    </form>
@endforeach

```

```
//route
```

```
Route::get('/Edit/pro_id={pro_id}',[ProductsController::class,'fetch_id']);
Route::post('/update',[ProductsController::class,'update']);
```

```
// delete
// Controller
```

```
public function del($pro_id){
    Product::where('pro_id',$pro_id)->delete();
    return redirect()->action([ProductsController::class,'getdata']);
}
```

```
// route
```

```
Route::get('/Delete/pro_id={pro_id}',[ProductsController::class,'del']);
```

Some more commands;

- ➔ To rollback the latest migration run the following command;
php artisan migrate:rollback
- ➔ To rollback a specific migration run the following command;
php artisan migrate:rollback --step=5
- ➔ If you have made some changes in more than one migrations and want those changes to be implemented run the following command;
php artisan migrate:refresh
- ➔ If you have made some changes in a specific migration and want those changes to be implemented run the following command;
php artisan migrate:refresh --step=5
- ➔ To drop all the table of a database and re-create them run the following command;
php artisan migrate:fresh --seed