Framework :-

It is the collection of methods, classes or files.

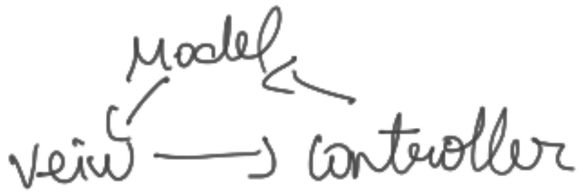
Architecture :-

It is a specific design a framework follows

what is laravel ?

Laravel is a PHP framework that uses MVC Architecture

MVC architecture :-

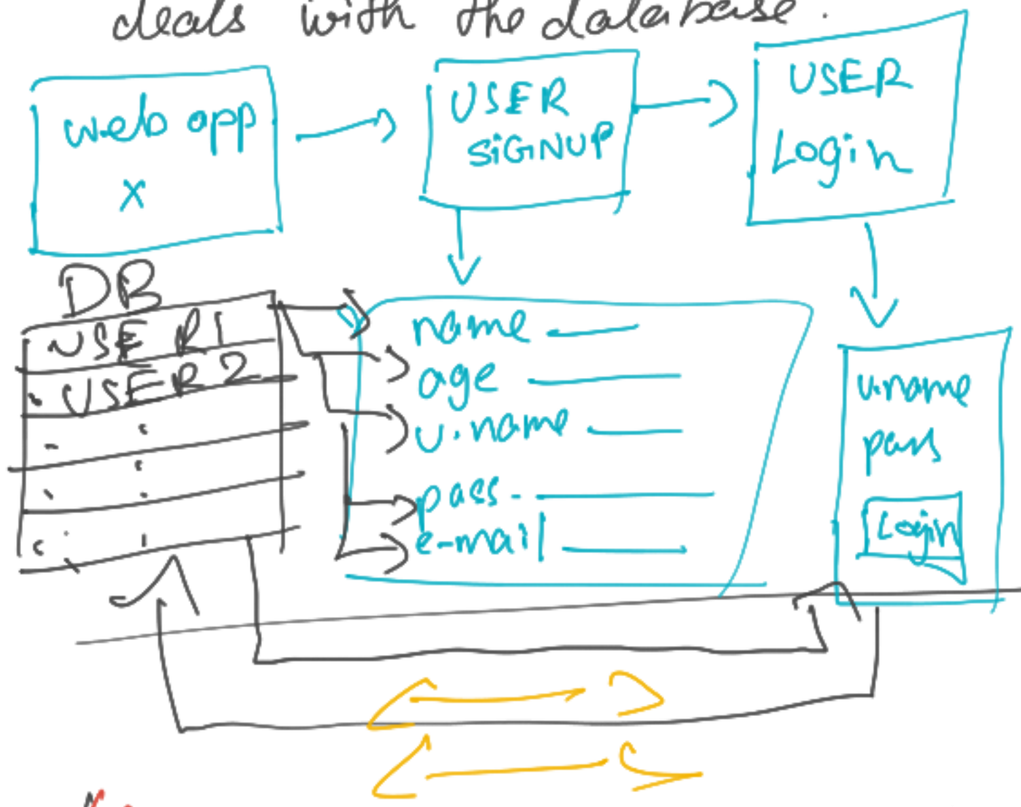


M → model

V → View

C → Controller

Model :- It is a class that deals with the database.



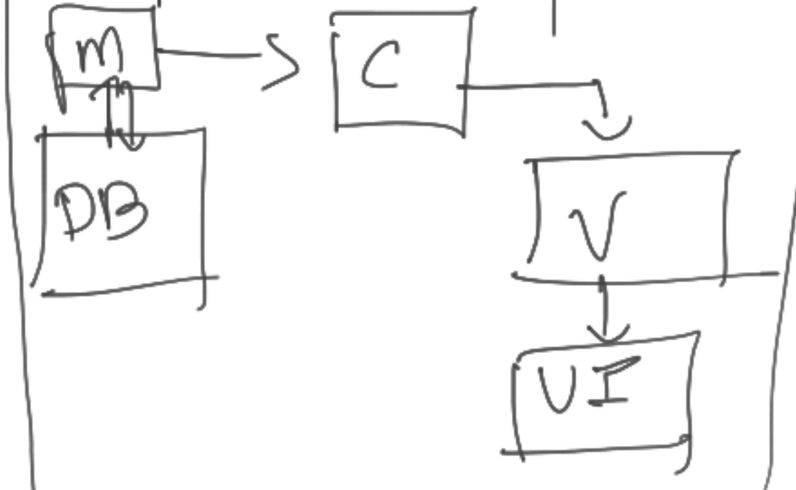
View -> A view is a class that deals with an HTML. Everything that can seen on the app! in the browser is view of the app.

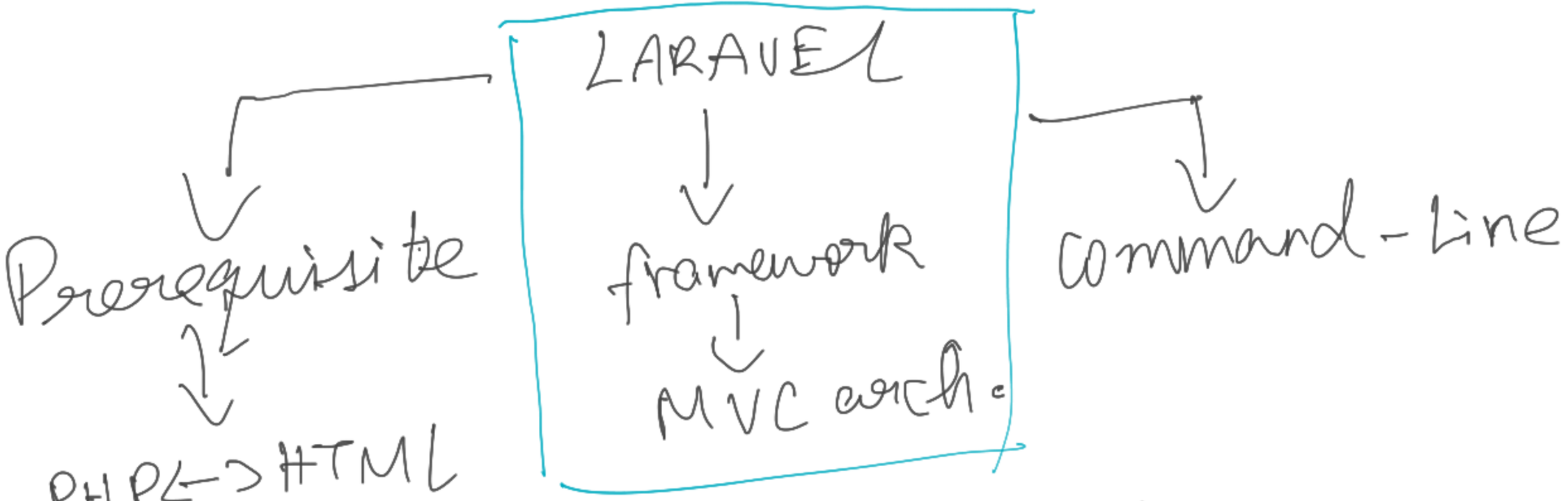
USER model

model -> its own table in DB.

Controller :-

A middleman b/w model & view





(1) Artisan → built in tool for a command-line known as Artisan.

↳ repetative programming tasks.

↳ used to create the skeleton code.

↳ database structure

↳ and their migration

↳ MVC files through the command line

↳ allows the dev. to create their own commands.

blade boilerplate

Prerequisites :→ you need.

↳ for your comp. to have

1. XAMPP →
2. Composer →

→ practical lab

PHP.

↳ composer.

Composer → dependency manager for PHP.

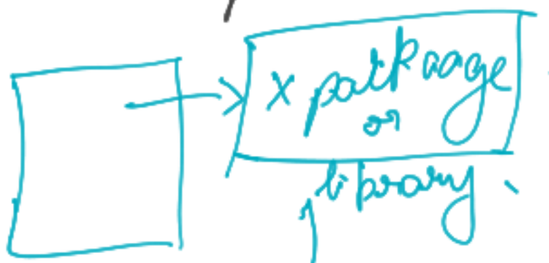
→ manages all the dep. of PHP software and req. lib.

It runs on command line → Artisan

Basic commands in composer

↳ require

```
{
  "require":
  {"<package name>":"ver:*"}
}
```



as a param.

↳ composer.json

↳ remove

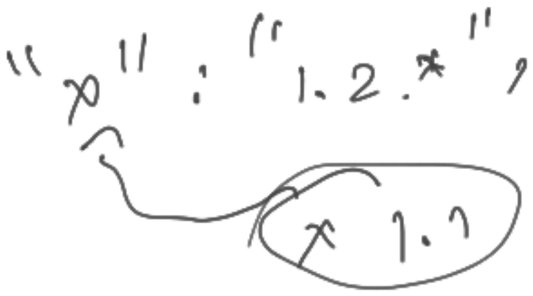
It is used to uninstall the lib. and remove it from the composer.json

↳ install

It is used to install all the lib. from the composer.json file.

↳ update

It is used to update all the lib. from composer.json. to the latest ver!



Creating your first Laravel project.

1. create locally
   - composer create-project/<project-name>
2. Create a global project
   - Composer global require laravel/installer
     - laravel new <project-name> Ex -> "y-app"

To start the local laravel server:-

y-app

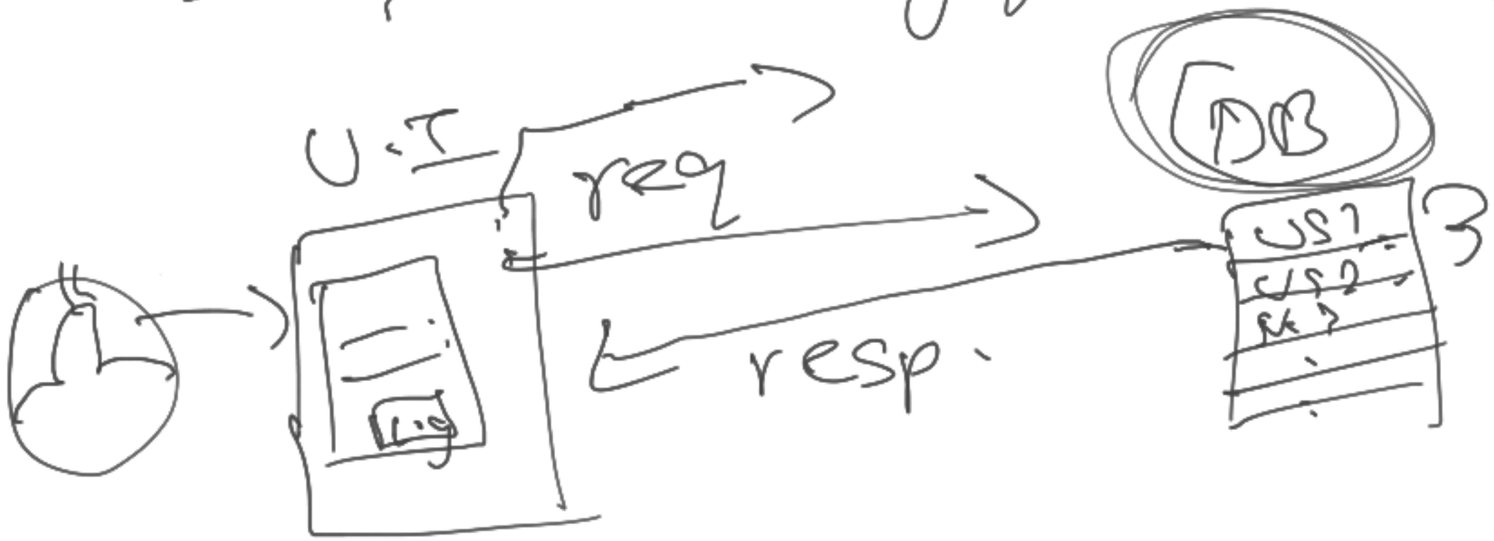
1. cd <project-name>
2. php artisan serve. (http://localhost:8000)

Request —> requiring something

↳ asking for something

Response —> responding to the req.

↳ answering for something



Routing:-

(MVC)

controller

The main functionality of the router is to route all your application requests to the appropriate controller.

By default

web.php :-> definition of default routes.

2-> The default routes are the routes of default web interface

api.php

↳ assigned with the API middleware group.

web.php (inside router directory)

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

```
Route::get('/example',
                function()
{ return "welcome";
});
```

login → app ⇄ DB

Hacker

credentials
U.N.
pass

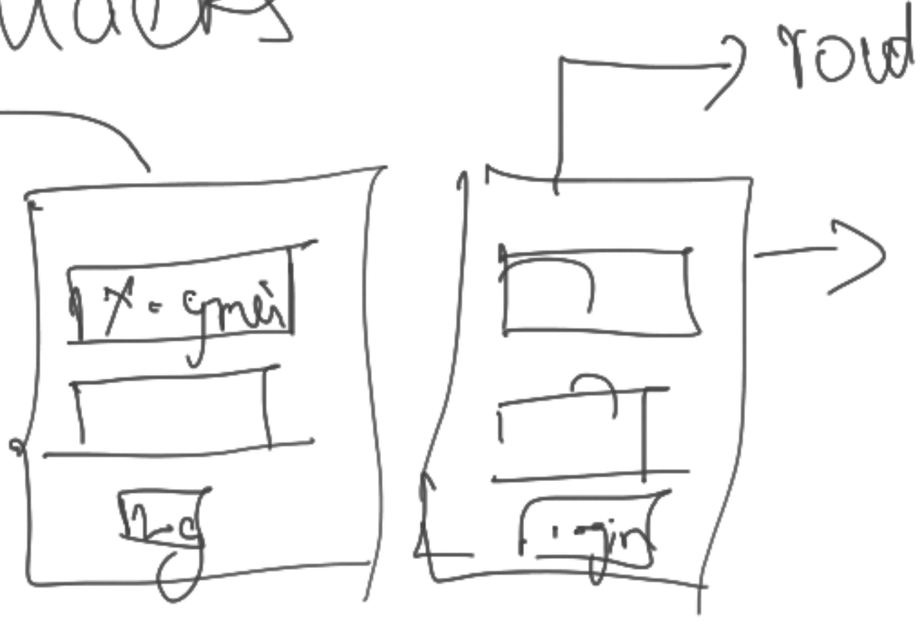
x.gmail.com
🔒

Cross site request forgery.

CSRF attacks

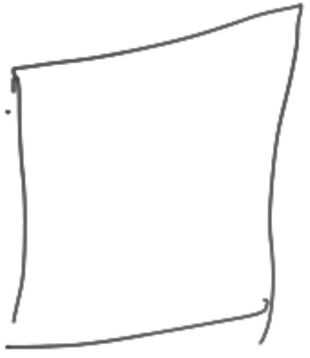
x-gmail.com

POST



POST, PUT, PATCH or DELETE ⟨ CSRF req.

100, 1000 ✓

In order to prevent CSRF req.,
Laravel automatically generates
CSRF tokens ⟩ token → 

USER 1

@csrf

Required Parameters :-


passing a data -> param. directly in the URL.

Required Parameters.

```
get("/", function ($id)
{
```

Optional Parameters :-

It is used where you want to make the route param. optional. (i.e passing it occassionally).

data -> URL 8000/user/KRISH

USER Name = KRISH

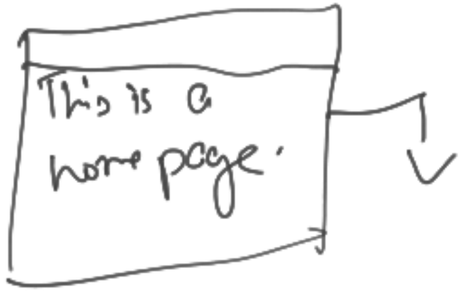
redirect()

view()

Parameter for Routing in Laravel :-

* Required param.
* Optional param.

```php
<?php
Route::get('/', function()
{
    return "This is a home page";
});
```



localhost:3000/laravel project/public/

-> /contact

localhost:3000/laravel projects/public/contact


```php
Route::get('/contact', function()
{
    return "This a contact page";
});
```

```
Route::get('/post/{id}', function($id)
{
    return "id number is:". $id;
}
);
```

required param.

null

localhost:3000/laravel projects/public/post

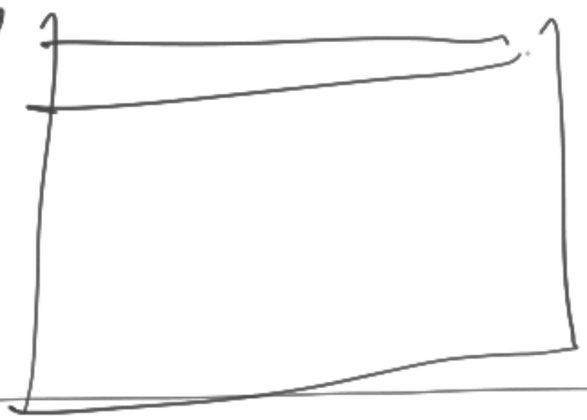
ERROR page not found

? Route::get('/post/{name?}', $name = NULL,
{return $name;
});

localhost/laravel/projects/public/post/

~~madhura~~
Rahul

Rahul

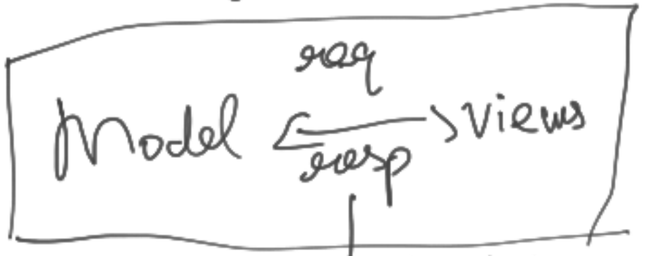


Controllers:- The middleman b/w the model & view.

↳ used to handle request logics.

RunQuery()

↳ app/Http/Controllers.



namespace App\Http\functions1;

" " " functions2;

"use"

How to create a controller?

STEP1 : Open cmd prompt in the project directory

STEP2 :

php artisan make:Controller <C.N>

class PostController extends Controller

{

//

}

PostController.php

Basic operations

-> C R U D

C → create
R → READ
U → update
D → delete

Boiler plate :- is a basic structure of a code.

php artisan make:controller --resource <C.N>

PostController

postController.php

create(), store(), show()
edit(), update(), destroy().

Routing Controllers:-

postController | web.php

```
Route::('/post', 'PostController@index');
```

postController.php | basic B-P

```
public function index()
{
    return "This page is awesome";
}
```

localhost:3000/laravel/project/public/post

How to pass data to the controller class?

postController.php

```
public function index($id)
{
    return "ID is:". $id;
}
```

web.php

```
Route::get('/post/{id}', postController@index');
```

localhost /laravelproject/post/50
{$id}

ID is: 50

Single Action Controllers

- use to perform single actions
  - has only one function.

__invoke()

```
Route::get('post/{id}', 'PostController');
```

post/68

id is: 68

Exceptions is occured

Unexpected value Expression

```
class PostController
      extends Controller
{ public function __invoke ($id)
  { retur "id is: ". $id;
  }
}
```

Template inheritance in Laravel :-

Master page layout :-

/resources/views/layouts/

create a new file named

↳ master.blade.php.

```
<html>
<head>
<title> Master page layout </title>
</head>
<body>
<div class="container">
@yield('content')
</div>
@yield('footer')
</body>
</html>.
```

@yield

↳ display Something.

Blade template :-

↳ templating engine

php A template created by blade.

↳ templating engine.

.php

.blade.php.

↳ blade template

X old php code

contact.blade.php

```
@extends('layout.master')
@section('content')
<h1> Contact page </h1>
@stop.
```

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