**Symfony Application**

-----------------------------------------------

Symfony Application

-----------------------------------------------

* Create Symfony Application - composer create-project symfony/website-skeleton my\_project\_name
* To start server - symfony serve
* then install mongodb from - composer require mongodb/mongodb
* ----composer require alcaeus/mongo-php-adapter
* composer config extra.symfony.allow-contrib true
* composer require doctrine/mongodb-odm-bundle
* if it is giving cache problem then add "doctrine/mongodb-odm-bundle": "^4.2", in composer.json and the update composer.
* start mongodb server
* Change in .ENV (Database name and Sqlite Driver)
* create Document.php
* create Controller and trial function
* create Type file in src/form/type

{ - Create src/form/Model/registation.php - to use userType data }

{ - Create Type file for registation with only UserType as field}

----------------------------------------------------------------NEXT NOTE---------------------------------------------------

To start server - symfony serve

To start and run server in background - symfony serve -d (then you can use same terminal for commands)

symfony plugins

For installing mongodb drivers - https://learnedia.com/install-mongodb-configure-php-xampp-windows/

then install mongodb from - composer require mongodb/mongodb

-----------------------------------------------

Annotation

--------------------

composer require annotations - use this to install

Writing routes in comment anouve the controller function is called annotations.

use /\*\* enter and write @Route("/") just before the functions.

-----------------

Twig

---------------------

composer require template

twig is use to use html template in symfony

------------------------------

Profiler

-----------

composer require profiler --dev

---------------------------------------------------

Public folder / app.css

-----------

create a css folder within public folder and create app.css file in that folder

---------------------------------------------------

----database--Object relational Mapper--

( https://symfony.com/doc/current/doctrine.html )

symfony console make:entity - for making table

dataabase creation

migration

controller creation

------------------

mongo install - https://stackoverflow.com/questions/20796714/how-do-i-start-mongo-db-from-windows

------------------

Mongodb steps ----------------

To create Collection and database in MongoDB - https://kb.objectrocket.com/mongo-db/show-collections-in-mongodb-1287#:~:text=To%20obtain%20a%20list%20of,least%20one%20collection%20is%20stored.

select database - use employee

create collection and insert data - db.employeeprofile .insert({

emp\_name: "Pradeep",

emp\_age: 23,

emp\_website: "employeeprofile.com"

})

Check the O/P - db.employeeprofile.find().pretty();

-------------------------------------------

Referencing in MongoDB -

https://www.slideshare.net/jwage/symfony-day-2010-doctrine-mongodb-odm?qid=def2d387-98d9-45f8-a3f8-873217f7c74a&v=&b=&from\_search=3

---------------------------------

Form-----------

https://symfony.com/doc/current/forms.html

--------------------------

for frontend of forms - REGISTER FRONT https://symfony.com/doc/4.1/doctrine/registration\_form.html#register-html-twig

-------------------------------

Uploaded file - public folder https://symfony.com/doc/current/controller/upload\_file.html

------------------------------------

controller features

https://symfony.com/doc/current/controller.html#fetching-services

---------------------------------

session - https://symfony.com/doc/current/session.html

--------------------------------

MOngoDB Import -

mongoimport --db CSV --collection filedata --type csv --headerline --ignoreBlanks --file \Users\urvashih\Desktop\data.csv

(https://medium.com/analytics-vidhya/import-csv-file-into-mongodb-9b9b86582f34)

------------------------------------

REGISTRATION

https://symfony.com/doc/current/bundles/DoctrineMongoDBBundle/cookbook/registration\_form.html

-------------------------------

Login - ORM

https://symfony.com/doc/current/security/form\_login\_setup.html#finishing-the-login-form

https://symfony.com/doc/4.0/security/form\_login\_setup.html

----------------------------------

For fetching mongodb data -

https://stackoverflow.com/questions/66207902/symfony-5-cant-get-a-way-to-read-the-property-tipoinvitado-in-class

----------------------------------

ARTUICLE UPLOAD - https://symfonycasts.com/screencast/symfony-uploads/upload-in-form

------------------------------

Controller ShortCuts

https://symfony.com/blog/new-in-symfony-2-6-new-shortcut-methods-for-controllers

-----------------------

ROLES-

class User {

/\*\*

\* @ORM\Column(type="array")

\*/

private $roles ;

public function getRoles()

{

$roles = $this->roles;

var\_dump($roles);

if ($roles != NULL) {

return explode(" ",$roles);

}else {

return $this->roles;

}

}

public function setRoles($roles)

{

$this->roles = $roles;

}

UserType

->add('roles', ChoiceType::class, array(

'attr' => array(

'class' => 'form-control',

'value' => $options[0]['roles'],

'required' => false,

),

'multiple' => true,

'expanded' => true, // render check-boxes

'choices' => [

'admin' => 'ROLE\_ADMIN',

'user' => 'ROLE\_USER',

]

))

-------------------------------------------

C R U D

http://www.inanzzz.com/index.php/post/htk5/simple-doctrine-mongodb-crud-example-in-symfony

-------------­

<https://medium.com/q-software/symfony-5-the-rest-the-crud-and-the-swag-7430cb84cd5>

**FormLoginAuthenticator.php**

The getCredentials() fetches the email address and password from the request, getUser() uses that to load the user and then checkCredentials() checks if the password matches the user account that getUser() fetched. If it matches and the user exists, then onAuthenticationSuccess() is called, otherwise onAuthenticationFailure() is called. The getLoginUrl() and getDefaultSuccessRedirectUrl() must be implemented since we are inheriting the GuardAuthenticator. These methods return the login URL and the URL that a successful login should redirect to respectively.The remember\_me functionality is not needed in this app, so we return false in supportsRememberMe().

CSV File Upload

• Create mongodb collection -----

• Create UI for uploading file ---

(https://codereviewvideos.com/course/how-to-import-a-csv-in-symfony)

• Read CSV, import and insert data from CSV to collection

• Fetch data using repository

• Update & delete data from collection

Article Site

• Create collections in mongodb

• UI for admin panel & user panel

• Register Users

• Login, Logout using sessions

• Update, delete user data

• CRUD for articles using image upload

• Add Comment to article & fetch it

• Add, Update, delete user using service

Symfony

- Open source PHP framework

- Collection of Tools which helps application to be organise, sable and scalable.

Benefits

-Enforce best practic (follow specific folder structure)

Concepts

-bundle(plugin)

-templates(twig)

-Services(dependency inje)

-Cache

DataBase with Doctrin

Fileformat for configuration

-Annotation

-YML

-XML

-PHP

Web Applivcation

-s/w that works on internet

-access with web browser

-Client side (html,css,javascript)

-Server side (PP/Symfony)

-Database(mysql)

Symfony Command

--Symfony Binary----

-symfony new my\_project --full (Web App)

-symfony new my\_project (microservices)

----Composer----

-composer create-project symfony/website-skeleton my-project-name (Full Web App)

-composer create-project symfony/skeleton my-project-name (microservices, api)

Running Symfony Application

-install server like Nginx or apache

Installing Packages/bundles

-provides ready to use features

-Symfony Flex(a tool to simplify the installation/removal of packages)

Symfony Packs (which are composer metapackages that includes several dependencies)

------------------ -----------------------------------------------NEXT NOTE--------------------------------------------------------**TechData Solution Store**

-Ecommerce : It is business model that lets firms and individual to bus and sell thinga over internet.

-SOSS : StreamOne Solution Store

-SOFA : StreamOne Frontend Application (used/provide product)

-Service Layer : It is a core business logic layer which handles order processing, provisioning process and future automated recurring billing functionality.

**CSS**

Relative Position - when the element is positioned relative to its normal position. (it will get change if we change the left right top in the property)

Absolute Position - The positioning is done relative to the first relatively (or absolutely) positioned parent element. In the case when there is no positioned parent element, it will be positioned related directly to the HTML element. (when you want to place elemet precisely where you want it. )

Fixed Position - It always stays in the same place even if the page is scrolled.

RWD stands for Responsive Web Design. This technique is used to display the designed page perfectly on every screen size and device, for example, mobile, tablet, desktop and laptop. You don't need to create a different page for each device.

MEDIA QUERY - The result of the query is true if the specified media type matches the type of device the document is being displayed on and all expressions in the media query are true. When a media query is true, the corresponding style sheet or style rules are applied.

The @mixin directive lets you create CSS code that is to be reused throughout the website.

The @include directive is created to let you use (include) the mixin.

Display

block - Element start with new line and takes full width of page.

inline - Element not start in new line and takes only required spaces.

inline-block - It allows to set a width and height on the element, it doesnot add line - break after the element, so the element can sit next to other elements.

-------------Part 2--------------

Cascading Style Sheet - CSS

There are three methods to integrate CSS on web pages.

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

<p style="color:blue">Hello CSS</p>

2) Internal/Embedded CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

<style>

p{color:blue}

</style>

3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

p{color:blue}

You need to link this style.css file to your html pages like this:

<link rel="stylesheet" type="text/css" href="style.css">

The link tag must be used inside head section of html.

CSS selector

There are several different types of selectors in CSS: -

CSS Element Selector - p{ }

CSS Id Selector - #id1{ } - unique

CSS Class Selector - .class1 { } - reuseable

CSS Universal Selector - \* { }

CSS Group Selector - h1 { }

Some CSS Style components are:

Selector

Property

Value

The CSS opacity property is used to specify the transparency of an element.

The CSS box model is used to define the design and layout of elements of CSS.

The elements are:

Margin - It removes the area around the border. It is transparent.

Border - It represents the area around the padding

Padding - It removes the area around the content. It is transparent.

Content - It represents the content like text, images, etc.

The z-index helps to specify the stack order of positioned elements that may overlap one another. The z-index default value is zero and can take on either a positive or negative number.

An element with a higher z-index is always stacked above than a lower index.

Z-Index can take the following values:

Auto: Sets the stack order equal to its parents.

Number: Sets the stack order of the element.

Initial: Sets this property to its default value (0).

Inherit: Inherits this property from its parent element.

visibility: hidden hides the element, but it occupies space and affects the layout of the document.7

display: none also hides the element but not occupy space. It will not affect the layout of the document.

W3C stands for World Wide Web Consortium. Its purpose is to deliver the information of the World Wide Web. It also develops rules and guidelines for the Web.

PSeudo element

::first-line

::before

::after

Relative Length

UNIT DESCRIPTION

em Relative to the font-size of the element (2em means 2 times the size of the current font)

ex Relative to the x-height of the current font (rarely used)

ch Relative to the width of the “0” (zero)

rem Relative to font-size of the root element

vw Relative to 1% of the width of the viewport\*

vh Relative to 1% of the height of the viewport\*

vmin Relative to 1% of viewport’s\* smaller dimension

vmax Relative to 1% of viewport’s\* larger dimension

% Relative to the parent element

Absolute Length

UNIT DESCRIPTION

CM centimetres

MM millimetres

IN inches (1in = 96px = 2.54cm)

PX pixels (1px = 1/96th of 1in)

PT points (1pt = 1/72 of 1in)

PC picas (1pc = 12 pt)

**HTML**

Layout of HTML

Following are different HTML5 elements which are used to define the different parts of a webpage.

<header>: It is used to define a header for a document or a section.

<nav>: It is used to define a container for navigation links

<section>: It is used to define a section in a document

<article>: It is used to define an independent, self-contained article

<aside>: It is used to define content aside from the content (like a sidebar)

<footer>: It is used to define a footer for a document or a section

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame.

Marquee is used to put the scrolling text on a web page. It scrolls the image or text up, down, left or right automatically. You should put the text which you want to scroll within the <marquee>......</marquee> tag.

Canvas element is used to draw graphics, on the fly, via JavaScript. The canvas element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

HTML 5 supports <video> tag and <audio> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

three types of video format:

mp4

WebM

Ogg

Html audio tag is used to add sound or music files on the web page. There are three supported file formats for HTML 5 audio tag.

mp3

WAV

Ogg

The HTML 5 datalist tag provides an autocomplete feature on the form element. It facilitates users to choose the predefined options to the users to select data. HTML

Layout of HTML

Following are different HTML5 elements which are used to define the different parts of a webpage.

<header>: It is used to define a header for a document or a section.

<nav>: It is used to define a container for navigation links

<section>: It is used to define a section in a document

<article>: It is used to define an independent, self-contained article

<aside>: It is used to define content aside from the content (like a sidebar)

<footer>: It is used to define a footer for a document or a section

The HTML iframe tag is used to display a nested webpage. In other words, it represents a webpage within a webpage. The HTML <iframe> tag defines an inline frame.

Marquee is used to put the scrolling text on a web page. It scrolls the image or text up, down, left or right automatically. You should put the text which you want to scroll within the <marquee>......</marquee> tag.

Canvas element is used to draw graphics, on the fly, via JavaScript. The canvas element is only a container for graphics. You must use JavaScript to actually draw the graphics.

Canvas is rendered pixel by pixel. In canvas, once the graphic is drawn, it is forgotten by the browser. If its position should be changed, the entire scene needs to be redrawn, including any objects that might have been covered by the graphic.

SVG is XML based, which means that every element is available within the SVG DOM. You can attach JavaScript event handlers for an element.

In SVG, each drawn shape is remembered as an object. If attributes of an SVG object are changed, the browser can automatically re-render the shape.

HTML 5 supports <video> tag and <audio> tag also. The HTML video tag is used for streaming video files such as a movie clip, song clip on the web page.

three types of video format:

mp4

WebM

Ogg

Html audio tag is used to add sound or music files on the web page. There are three supported file formats for HTML 5 audio tag.

mp3

WAV

Ogg

The HTML 5 datalist tag provides an autocomplete feature on the form element. It facilitates users to choose the predefined options to the users to select data.

**JAVASCRIPT**

It is widely used for client-side validation.

The types of function are:

Named - These type of functions contains name at the time of definition. For Example:

function display()

{ document.writeln("Named Function");

}

display();

Anonymous - These type of functions doesn't contain any name. They are declared dynamically at runtime.

var display=function()

{ document.writeln("Anonymous Function");

}

display();

argument object

The variables of JavaScript represent the arguments that are passed to a function.

3 Places to put JavaScript code

Between the body tag of html

<script type="text/javascript">

alert("Hello Javatpoint");

</script>

Between the head tag of html

<head>

<script type="text/javascript">

function msg(){

alert("Hello Javatpoint");

}

</script>

</head>

In .js file (external javaScript)

message.js----

function msg(){

alert("Hello Javatpoint");

}

message.html-----

<head>

<script type="text/javascript" src="message.js"></script>

</head>

BOM stands for Browser Object Model. It provides interaction with the browser. The default object of a browser is a window. So, you can call all the functions of the window by specifying the window or directly. The window object provides various properties like document, history, screen, navigator, location, innerHeight, innerWidth.

DOM stands for Document Object Model. A document object represents the HTML document. It can be used to access and change the content of HTML.

The window object is created automatically by the browser that represents a window of a browser. It is not an object of JavaScript. It is a browser object.

The window object is used to display the popup dialog box. Let's see with description.

alert() - displays the alert box containing the message with ok button.

confirm() - displays the confirm dialog box containing the message with ok and cancel button.

The history object of a browser can be used to switch to history pages such as back and forward from the current page or another page. There are three methods of history object.

history.back() - It loads the previous page.

history.forward() - It loads the next page.

history.go(number) - The number may be positive for forward, negative for backward. It loads the given page number.

The == operator checks equality only whereas === checks equality, and data type, i.e., a value must be of the same type.

There are 3 ways to create an object in JavaScript.

By object literal

By creating an instance of Object

By Object Constructor

Client-side JavaScript comprises the basic language and predefined objects which are relevant to running JavaScript in a browser. The client-side JavaScript is embedded directly by in the HTML pages. The browser interprets this script at runtime.

Server-side JavaScript also resembles client-side JavaScript. It has a relevant JavaScript which is to run in a server. The server-side JavaScript are deployed only after compilation.

----------------------------------------------------------------INTERVIEW----------------------------------------------------------

Multiple not supported by php.

a + b

||

c

Multi level is supported.

a

|

b

|

c

Explain the difference between GET and POST requests.(HTTP Protocols)

•GET allows displaying the submitted data as part of the URL. This is not the case when using POST as during this time, the data is encoded in the request.

•Typically, GET is used for retrieving data while POST is used for inserting and updating data.

PUT - Update and Replace

PATCH - Update and Modify

What are the Traits in PHP?

The mechanism allows for the creation of reusable code in PHP-like languages where there is no support for multiple inheritances. A trait can’t be instantiated on its own.

PHP Implode Function - array to string

-The implode function in PHP is used to "join elements of an array with a string".

-The implode() function returns a string from elements of an array. It takes an array of strings and joins them together into one string using a delimiter (string to be used between the pieces) of your choice.

PHP Explode Function - string into an array

-The explode function is used to "Split a string by a specified string into pieces i.e. it breaks a string into an array".

-The explode function in PHP allows us to break a string into smaller text with each break occurring at the same symbol. This symbol is known as the delimiter.

What are Magic Methods in PHP?

Methods that begin with 2 underscores(\_\_) are generally called Magic methods in PHP. These methods names are limited to some list of php supported keywords that are reserved. So any function should not be defined with the name of PHP magic methods.

Usually, these functions should be defined by the user and there is no need to call them explicitly.

List of Magic Methods in PHP

• \_\_construct()

• \_\_destruct()

• \_\_call($fun, $arg)

• \_\_get($property)

• \_\_set($property, $value)

• \_\_isset($content)

• \_\_unset($content)

isset() function checks if the particular variable is set and has a value other than NULL.

Checked exception is the exception that occurs at the compile time. As it is not possible to ignore this type of exception, it needs to be handled cautiously.

An unchecked exception, on the other side, is the one that occurs during the runtime. If a checked exception is not handled, it becomes an unchecked exception.

**ANGULAR**

AngularJS uses MVC or Model-View-Controller architecture, where the Model contains the business logic, Controller processes information and View shows the information present in the Model.

Angular replaces controllers with Components. Components are nothing but directives with a predefined template.

In Angular, property binding is done using "[ ]" attribute and event binding is done using "( )" attribute.

Just in Time Compiler (JIT) and Ahead of time Compiler (AOT)

In JIT compilation, the application compiles inside the browser during runtime.

Whereas in the AOT compilation, the application compiles during the build time.

ANGULAR APP----

- ng new angularApp

Components : ng generate component name

- components are the basic building blocks, which control a part of the UI for any application.

- Define using @Component decorator.

- Every component consists of three parts, the

+ template which loads the view for the component,

+ a stylesheet which defines the look and feel for the component,

+ and a class that contains the business logic for the component.

Modules - ng g m name

- A module is a place where we can group components, directives, services, and pipes.

- Module decides whether the components, directives, etc can be used by other modules, by exporting or hiding these elements.

- defined with a @NgModule decorator.

Services - ng g s name

- Services are objects which get instantiated only once during the lifetime of an application.

- The main objective of a service is to share data, functions with different components of an Angular application.

- A service is defined using a @Injectable decorator.

-Any method/function defined inside the Service class can be directly used inside any component by just importing the service.

lifecycle hooks in Angular

ngOnChanges() - whenever one or more input properties of the component changes.

ngOnInit() - It initializes the component and sets the input properties of the component.

ngDoCheck() - It is used to detect and act on changes that cannot be detected by Angular.

ngOnDestroy() - It gets called just before Angular destroys the component. This hook can be used to clean up the code and detach event handlers.

Data Binding - a way to communicate between the component(Model) and its view(HTML template).

- String interpolation and property binding allow only one-way data binding.

-String interpolation uses the double curly braces {{ }} to display data from the component.

-Using property binding, we can bind the DOM properties of an HTML element to a component's property. Property binding uses the square brackets [ ] syntax.

Pipes/Filters

Angular uses pipes which can be used to format data before displaying it. ex: lowercase, uppercase, Currency converter.

Observables

- Emit multiple values over a period of time.

- Are lazy, they’re not executed until we subscribe to them using the subscribe() method.

- have unsubscribe() method.

- Deliver errors to the subscribers.

Promises

- Emit a single value at a time.

- Are not lazy: execute immediately after creation.

- Are not cancellable.

Directives

- A directive is a class in Angular that is declared with a @Directive decorator.

- Every directive has its own behaviour and can be imported into various components of an application.

- Use where multiple components need to have similar functionalities. In such a situation, one can create a directive having the required functionality and then, import the directive to components which require this functionality.

- Types of directives

+ Component directives - @component

+ Structural directives -

- \*ngIf is used to check a boolean value and if it’s truthy,the div element will be displayed.

- \*ngFor is used to iterate over a list and display each item of the list.

+ Attribute Directives - ng g directive name

-These directives are used to change the look and behaviour of a DOM element.

share data between components in Angular

- Parent to child using @Input decorator

selector: 'app-parent',

template: `

<app-child [data]=data></app-child>

` ,

export class ParentComponent{

data:string = "Message from parent";

----------------------

selector: 'app-child',

template:`

<p>{{data}}</p>

`,

export class ChildComponent {

@Input() data:string

- Child to parent using @ViewChild decorator

Dependency Injection

- Dependencies in angular are nothing but services which have a functionality. Angular provides a smooth mechanism by which we can inject these dependencies in our components and directives.

- we can create injectable dependencies by adding the @Injectable decorator to a class.

-------------PART 2----------------

ANGULAR

User Services(provide URLs)

Validator(Confirm Password and Password)

Validations - errors,touched,dirty,required,min,max

ngOnInit(to check if user is already loggedin or not)

AuthGuard (To check user is login or not)

Reactive Forms- FormBuilder Service

Behaviour subject

FLOW in Frontend

UI layer(HTML Page)-ngSubmit=register() on submit Event ->

Controller(.ts file)-register function is describe and userservice is subscribed and called ->

Service Layer(userService) -User service have a function register which have parameter which takes all the value of form and it attatched it with api URL and send data to backend.

In Backend

In api.php post route is set to take all the incoming data into the User controller, in usercontroller register function is describe it will take all the data and send it to data base using create function.

**TRAINING PROJECT**

Mini Project - Blog It --- CSS, HTML, Bootstrap,Written JSON Api

- It is a blogging website where user can post and read blogs of other users.

-User can add photos in the blog

-User can also search blogs according to different catagory or by words.

-For writing the blog user have given the text editor with features of different fonts and different sizes.

-User can see the already written blogs in carousel.

-We also add the functnality of comment.

-User can read the some part of the blogs without getting signup, but for reading the whole blog user have to login.

-For login and signup i use modals.

Major Project - ShoutBox --- Apache server, Mysql Database, Laravel Framework, Angular 8, Execute API using PostMan

-We have created a site where user can post images, video, and text.

-User can make new friends by accepting and sending the request.

-User can add or update there details in profile.

-User can see the post of friends and react on them by liking or commenting on them.

-User can also report the post of other user.

ADMIN

-There was 2 module Admin and User.

-Admin have to accept the request of the new user then only user can login into the website.

-Admin can delete the reported Post and Users.

-Admin can see all the Users and their posts.

Database

-We created Single table for user having column for role Admin and User, in controller logic using if condition i was checking if the user is admin or not.

VALIDATION

-required, minimum,maximum,email,password

What I did ?

-My part was doing front end and back end of Registration and login

-I made 2 componets for Login and signup in angular and user controller and user Model in backend laravel.

-In backend we were using Laravel Eloquent ORM (Object Realation Mapper) Ex: if we make table name users then the model name will be user.

-In user controller i wrote query using Query builder for fetching data and send that data in Json api.

Api

-We were using api to connect front end with backend, so i made user services where i was providing the url of api and from there data will be fetched from database

Approach new Technology

Studing from basiscs

study from udemy courses

practice and doing hands-on

Optimize the code

sonarscanner and test cases.

**JOINs**

Joins

-Inner Join

-Let Join

-Right Join

-Outer Join

(INNER) JOIN: Returns records that have matching values in both tables

LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table

RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table

FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

**Laravel Eloquent**

Eloquent ORM (Object Realation Mapper) provides an easy way to communicate with Database.

Can work with multiple databases efficiently.

It is an architrctural pattern where the model created in the MVC structure corresponds to a table in the database.

Ex: if we make table name users then the model name will be user.

Using Eloquent ORM Model can perform common queries and can insert data into the database using seeder.

All we have to do is to define the database tables and relation between them, and Eloquent will do the rest.

We can perform CRUD operations with the Eloquent ORM

1. Create - ::create()

2. Retriving - ::all() ::find() ::where()->first() ::where()->get()

3. Updating - find -> update -> save()

4. Deleting - find->delete() find->destroy()

MVC -

Model : Stores and Manages data

View : User Interface (Visual Representation of data, user direct interact with view)

Controller : Business Logic, it connect the model and view (The controller converts inputs from the view to retrieve/update data in the model)

Flow

view -> controller -> Model

|

view <- Controller <-

OVER LOADING & OVERRIDING

Overloading : When two or more methods in the same class have the same name but different parameters, it's called Overloading or Compile Time polymor.

Overriding : When the method signature (name and parameters) are the same in the superclass and the child class, it's called Overriding or Run Time Polymor.

Inhiritance : One class is allowed to inherit the features(fields and methods) of another class.

BASE CLASS ->extends DERIVED CLASS

**Use of .htaccess and php.ini files in PHP?**

Both of them are used for making changes to the PHP settings.

.htaccess – A special file that can be used to change or manage the behavior of a website. Directing all users to one page and redirecting the domain’s page to https or www are two of the most important uses of the file. For .htaccess to work, PHP needs to be installed as an Apache module.

php.ini – This special file allows making changes to the default PHP settings. Either the default php.ini file can be edited, or a new file can be created with relevant additions and then saved as the php.ini file. For php.ini to work, PHP needs to run as CGI.