
	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

/

Question Bank

S#	Question	Marks	Unit	R#	Verifier R#
1	What is laravel.(1 M) and what is the laravel framework architecture?	1 M 5 M	1	A-1	A-22
2	What are features of laravel?	5/7	1	A-2	A-23
3	How do you install a laravel project on a PC?	1 M	1	A-3	A-24
4	How do you configure mysql environment using environment file(env file)	1 M	1	A-4	A-25
5	Describe all types of artisan command in laravel with proper explanation.	5/7 M	1	A-5	A-26
6	Explain basic Route in laravel.	5/7 M	1	A-6	A-27
7	Explain available router methods in laravel.	5/7 M	1	A-7	A-28
8	Explain Route redirect in laravel	1 M	1	A-8	A-29
9	Explain Route parameters in laravel.	5/7 M	1	A-9	A-30
10	Explain Route Regular Expression Constraints.	5/7 M	1	A-10	A-31
11	Explain Named Routes in laravel.	1/5 M	1	A-11	A-32
12	Explain how to create a controller in laravel.	1 M	2	A-12	A-33
13	Explain Resource Controller in laravel(update,show,destroy,index,store,create,edit-7 Methods).	5/7 M	2	A-13	A-34
14	Explain How to pass data variable from controller to view with example and explanation?	5/7 M	2	A-14	A-35
15	Explain Route Model Binding.	5/7 M	2	A-15	A-36
16	Explain CSRF token.	1 M	2	A-16	A-37
17	Explain request Headers OR explain method facking in laravel.	1 M	2	A-17	A-38
18	Explain responses in view.	5/7 M	2	A-18	A-39

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

S#	Question	Marks	Unit	R#	Verifier R#
19	Explain @yield, @section, @include, @Stack, @push, @if, @if-empty, @foreach, @forelse, @while, @switch (blade_directories)	5/7 M	3	A-19	A-40
20	Explain json responses in laravel.	1 M	2	A-20	A-41
21	What are blade templates and explain blade templates in laravel?	5/7 M	3	A-21	A-42


Q.1. Question [Name: ABC, Enroll: 123, Class: A, Roll: 098]

-

Rules

1. Fo

-

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q-1.What is laravel.(1 M) and what is the laravel framework architecture?

[Name: Mansee Adhiya, Enrollment: 230823001, Class: A, Roll: 01]

Verified By: [Name: Vishakha Kasundra, Enrollment: 230823105, Class: A, Roll: 22]

Ans:-

Laravel is a free, open-source web framework based on PHP.

Laravel Framework Architecture:

1.Model-View-Controller (MVC) Pattern:

Laravel follows the MVC architecture, where models represent data, views display the UI, and controllers handle user requests and responses.

2.Routing:

Laravel's routing system directs incoming HTTP requests to the appropriate controller methods, allowing for clean and organized code structure.

3.Eloquent ORM:


Laravel's ORM (Object-Relational Mapping) simplifies database interactions by providing a fluent, expressive syntax for defining database relationships and executing queries.

4.Blade Templating Engine:

Laravel's Blade templating engine allows for the creation of reusable, modular views with intuitive syntax, enabling efficient UI development.

5.Middleware:

Middleware in Laravel provides a mechanism to filter HTTP requests entering the application. It's useful for authentication, authorization, and other request preprocessing tasks, enhancing security and control over application flow.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.2. What are features of laravel?

[Name: Krupali Akabri, Enrollment: 230823003, Class: A, Roll: 02]

[Verified by: Khan Shaban S. Enroll:230823053 Class-A RollNo:23]

ANS.

Introduction:

Laravel is a well-known open-source PHP web framework praised for its elegant syntax and developer-friendly features. It streamlines the creation of trustworthy and scalable web applications by providing various tools, libraries, and conventions. Laravel features include a strong routing system, an expressive FORM (Object-Relational Mapping) for database interactions, built-in authentication systems, and a template engine called Blade with a focus on the readability and maintainability of the code.

Features of Laravel:

1. The Innovative Template Engine

The pre-made, lightweight templates of the Laravel framework are recognized for enabling the creation of attractive layouts with dynamic content seeding. It also offers a variety of necessary widgets that combine CSS and JS code using solid frameworks. With little developer effort, the meticulously crafted Laravel templates enable the creation of simple and composite layouts.

2. MVC Architecture Support through Built-In Functions

The MVC design pattern, which Laravel supports, keeps the important display and business logic layers separate. The MVC design of Laravel has several built-in features that aid developers in building more efficient, scalable, and secure apps.


The permission and authentication section of the code is crucial when creating an application. It also estimates the time a programmer needs to create a new object. The framework also includes a logical approach that aids in regulating access to various resources and reforming authorization logic.

3. Authorization Technique for Coding

The amount of time spent creating has significantly lowered thanks to the validation of the Laravel-integrated computer code. Thanks to the application reasoning capability, developers have the independence and flexibility to work independently on software projects of all sizes.

4. Eloquent Object Relational Mapping and Implementation

Eloquent Object Relational Mapping (ORM) by Laravel has a simple PHP Active Record implementation. Web application developers can now create database queries using PHP syntax rather than SQL code. Compared to prior PHP frameworks, ORM is faster.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

5. Strong Application Security

One of Laravel's best advantages is its security measures. The Laravel framework gives developers access to advanced and robust online application security. The hashed and salted password methods used by Laravel's security features effectively prevent plain-text password storage in user databases.

6. Artisan Tool for Automating Programming Tasks

The Laravel framework's Artisan command-line tool aids in automating time-consuming and repetitive programming tasks. The artisan tool is also used to build the database's basic structure and coding. Managing various database systems is made simple by its usage in migration job management. Additionally, generating unique MVC files through the command line may communicate with those assets and their configurations. Artisans even assist developers in the creation and usage of commands.

7. Enable Libraries and Modular Functionalities

Laravel aids developers by providing pre-installed Object-Oriented and Modular libraries, which are absent from many other PHP frameworks. For instance, the Authentication library's implementation is simple and includes several functions.

These procedures include encryption, password reset, Bcrypt hashing, active user verification, and CSRF (Cross-site Request Forgery) protection. A responsive and modular web application may be created using this framework since it is separated into independent modules utilizing cutting-edge PHP ideas.

8. Facilitate Database Migration System through PHP Code

The Laravel migration mechanism allows it to expand the web application database structure rather than starting from scratch each time there is a code change. Data loss is extremely unlikely as a result of this feature.


The feature makes it feasible to alter the complete database structure by utilizing PHP code rather than SQL. Additionally, Laravel Schema Builder aids developers in building database tables and, if needed, inserts indexes with the required columns.

9. Competitive and Effortless Unit Testing

The finest framework for creating web applications is Laravel because it effectively supports unit testing features. This framework can perform many unit tests to ensure that new modifications made by the programmer do not unexpectedly break the current web application. Because Laravel is aware of any found issues, it may deliver updates on a frequent basis. Furthermore, it facilitates the developers' ability to create unit tests for the scenario.

10. Useful Tutorials

Laravel provides a source of learning for novice and seasoned developers through its [Laracast lessons](#). You can learn to use Laravel, whether looking at a free video lesson or a premium service. Understanding this technology framework is made simple by the precise, understandable, and uncomplicated learning instructions that may be used to construct and reproduce training materials.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q-3. How do you install a laravel project on your PC?

[Name: Janvi Ashara, Enrollment:230823004, Class:A, Roll:03]

Verified By: [Name: jay khunt, Enrollment: 230823054, Class: A, Roll: 24]

Ans:-

Via Composer create-project


You have type the **create-project** command in your Terminal:

Syntax:

composer create-project --prefer-dist

Example:

Composer create-project laravel/laravel myapp

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.4. How do you configure a mysql environment using environment file(env file)?


[Name : Bhalodiya Milan Dineshbhai, Enrollment No. : 230823008, Class : A, Roll No. : 04]

Verified By: [Name: krishna khunt, Enrollment: 230823055, Class: A, Roll No: 25]

Ans : .env file is used to connect the database with your project.

To connect with a database you have to make changes in the file where DB_DATABASE=Laravel, you will write your database name in place of Laravel.

As well as if you have changed your username and password you have to provide it in the section where DB_USERNAME=root DB_PASSWORD=, here you will provide your new username and password you have set.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.5.Describe all types of artisan command in laravel with proper explanation.

[Name: Drashti Chavda, Enrollment: 230823013, Class: A, Roll: 05]

Verified By: [Name: lilapara mukesh, Enrollment: 230823056, Class: A, Roll: 26]

Ans:

Introduction: Artisan is the name of the command-line interface included with Laravel. It provides a number of helpful commands for your use while developing your application. It is driven by the powerful Symfony Console component.

Listing All Available Commands:

To view a list of all available Artisan commands, you may use the `list` command:

`php artisan list`

Viewing The Help Screen For A Command:

Every command also includes a "help" screen which displays and describes the command's available arguments and options. To view a help screen, simply precede the name of the command with `help`:

`php artisan help migrate`

Specifying The Configuration Environment:


You may specify the configuration environment that should be used while running a command using the `--env` switch:

`php artisan migrate --env=local`

Displaying Your Current Laravel Version:

You may also view the current version of your Laravel installation using the `--version` option:

`php artisan --version`

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.6. Explain basic Route in laravel.

[Name: Hemangi Chavda, Enrollment:230823014, Class:A, Roll:06]

In Laravel, a route is a way to direct incoming HTTP requests to specific controllers or closures, allowing you to define how your application responds to different URLs. Routes are defined in the `routes` directory of a Laravel application.

There are two basic types of routes in Laravel:

Basic Routes: These routes respond to HTTP requests for specific URIs. Basic routes typically accept a URI and a closure, providing a very simple and expressive method of defining routes without the need for a controller. Here's an example of a basic route:

```
Route::get('/hello', function () {
    return 'Hello, World!';
});
```


In this example, when a GET request is made to the `/hello` URI, the closure function will be executed and it will return the string `'Hello, World!'`.

Route Parameters: Route parameters allow you to capture parts of the URI within your route definition. This allows you to create dynamic routes that can respond differently based on the values passed in the URI. Here's an example of a route with parameters:

```
Route::get('/user/{id}', function ($id) {
    return 'User ID: ' . $id;
});
```

In this example, the `{id}` portion of the route indicates a route parameter. When a request is made to a URL like `/user/123`, the value `123` will be passed as the `$id` parameter to the closure function, and it will return `'User ID: 123'`.

These are the basic concepts of defining routes in Laravel. Routes are typically defined in the `routes/web.php` file for web routes and `routes/api.php` file for API routes, although you can create additional route files if needed.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.7. Explain available router methods in laravel.

[Name: Neha Dobariya, Enroll: 230823020, Class: A, Roll: 7]

[Verified by: Sonali A. Mahidhariya Enroll:230823059 +Class-A RollNo:28]

Ans:

In Laravel, there are several types of routes you can define based on the HTTP methods and the purpose of the route. The main types of routes in Laravel are as follows:

1- Basic Routes: Basic routes are the most common type of routes in Laravel. They respond to HTTP requests like **GET, POST, PUT, DELETE**, etc., and map the URL to a specific controller method or closure function. For example:

// GET route

```
Route::get('/home', 'HomeController@index');
```

// POST route

```
Route::post('/submit', 'FormController@submit');
```

// PUT route

```
Route::put('/update/{id}', 'UserController@update');
```

// DELETE route

```
Route::delete('/delete/{id}', 'UserController@delete');
```

2- Route Parameters: You can define route parameters to capture parts of the URL and pass them as arguments to your controller methods. Parameters are enclosed in curly braces {}. For example:


```
Route::get('/user/{id}', 'UserController@show');
```

3- Named Routes: Named routes allow you to assign a unique name to a route. This makes it easier to reference the route in your application's code. You can use the **name** method to name a route. For example:

```
Route::get('/profile', 'ProfileController@show')->name('profile.show');
```

4- Route Groups: Route groups allow you to apply common attributes, such as middleware or a prefix, to a group of routes. This helps keep your routes organized and makes it easier to maintain the application. For example:

```
Route::prefix('admin')->middleware('auth')->group(function () {
    // Admin routes go here...
});
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

```
});
```

5- Resource Routes: Resource routes are used for creating routes that follow RESTful conventions. They automatically generate routes for common CRUD operations (Create, Read, Update, Delete) for a resource (e.g., posts, products) in a controller. For example:


```
Route::resource('posts', 'PostController');
```

6- API Resource Routes: API resource routes are similar to resource routes but are typically used for building API endpoints. They generate routes for standard CRUD operations but without the need for rendering views. For example:

```
Route::apiResource('products', 'ProductController');
```

7- Fallback Routes: Fallback routes are used to handle requests that don't match any of the defined routes. These routes are useful for implementing custom 404 error pages or handling unknown routes. For example:

```
Route::fallback(function () {
    // Custom 404 error page or response
});
```


	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.8 Explain Route redirect in laravel.

[Name:Gadara Vidheybhai D. , Enroll:230823022 , Class-A , Roll No.-08]

[Verified by:Makwana Vishal G. Enroll:230823062 Class-A RollNo:29]

Ans: Redirects are needed when users visit routes that they are not permitted to visit or try to perform actions that they are not permitted to perform. Laravel offers two common ways to handle redirects and send users to where they should be.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.9 Explain Route parameters in laravel.

[Name:Gautam Rahul F. , Enroll:230823023 , Class-A , Roll No.-09]

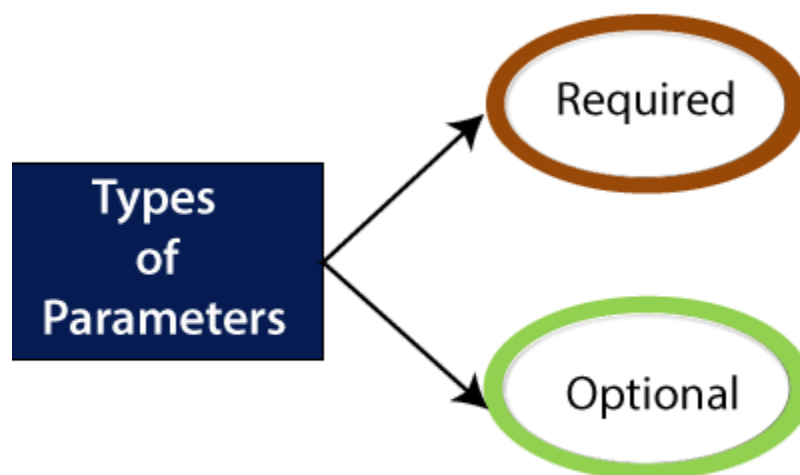
Laravel routes are located in the app/Http/routes.php file. A route usually has the URL path, and a handler function callback, which is usually a function written in a certain controller.


A parameter provided in the route is usually annotated with curly braces. For instance, to pass in a name parameter to a route, it would look like this.

```
Route::get('/params/{name}', function ($name) {
    return $name;
});
```

There are two types of parameters we can use:

- **Required Parameters**
- **Optional Parameters**



	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Required Parameters

The required parameters are the parameters that we pass in the URL. Sometimes you want to capture some segments of the URI then this can be done by passing the parameters to the URL. For example, you want to capture the user id from the URL.

```
Route::get('/post/{id}', function($id)
```

```
{ return "id number is : ".
```

```
$id; }
```

```
);
```

The route parameters are enclosed within {} brackets, and parameters must contain alphabetic characters. It should not contain '-' character, and instead of using this character, you can use '_' character.

Route parameters are available in the route callbacks. Syntax of route parameters is given below:


Name of the callback/controller arguments

Optional Parameters

Suppose you want to specify the route parameter occasionally, in order to achieve this, you can make the route parameter optional. To make the route parameter optional, you can place '?' operator after the parameter name. If you want to provide the optional parameter, and then make sure that you have also provided the default value to the variable.

```
Route::get('user/{name?}', function ($name=null) {
```

```
return $name;
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

});

we observe that the parameter we pass in the URL is optional. As we have provided the default value to the parameter as Null, so if we do not pass any parameter, it will return null. If we pass the parameter in the URL, then the value of the parameter would be displayed.

Q.10 Explain Route Regular Expression Constraints.

[Name: Harsoda Hetasvi R, Enroll:230823029,Class-A, RollNo:10]

[verified by: Marsonia Reesha D, Enroll:230823065, class-A RollNo:31]

Ans: You may constrain the format of your route parameters using the where method on a route instance. The where method accepts the name of the parameter and a regular expression defining how the parameter should be constrained.


Example:

```
Route::get('/user/{name}', function (string $name) {
    })->where('name', '[A-Za-z]+');
Route::get('/user/{id}', function (string $id) {
    })->where('id', '[0-9]+');
Route::get('/user/{id}/{name}', function (string $id, string $name) {
    })->where(['id' => '[0-9]+', 'name' => '[a-z]+']);
```

some commonly used regular expression patterns have helper methods that allow you to quickly add pattern constraints to your routes.

Example:

```
Route::get('/user/{id}/{name}', function (string $id, string $name) {
    })->whereNumber('id')->whereAlpha('name');
Route::get('/user/{name}', function (string $name) {
    })->whereAlphaNumeric('name');
Route::get('/user/{id}', function (string $id) {
    })->whereUuid('id');
Route::get('/user/{id}', function (string $id) {
    })->whereUlid('id');
Route::get('/category/{category}', function (string $category) {
    })->whereIn('category', ['movie', 'song', 'painting']);
```

	<p style="text-align: center;">ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science</p>	
	<p style="text-align: center;">23PGCA203-Core-Laravel For Modern Web Development</p>	<p style="text-align: center;">Rev. No: 001/Dec-2023</p>

Global Constraint:

If you would like a route parameter to always be constrained by a given regular expression, you may use the `pattern` method. You should define these patterns in the `boot` method of your `App\Providers\RouteServiceProvider` class.

Example:

```
public function boot(): void
{
    Route::pattern('id', '[0-9]+');
}
```

Once the pattern has been defined, it is automatically applied to all routes using that parameter name.

Example:

```
});


Route::get('/user/{id}', function (string $id) {
```

Enclosed Forward Slashes:

The Laravel routing component allows all characters except `/` to be present within route parameter values. You must explicitly allow `/` to be part of your placeholder using a `where` condition regular expression.

Example:

```
Route::get('/search/{search}', function (string $search) {
    return $search;
})->where('search', '.*');
```


	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.11.Explain Named Routes in laravel.

[Name: Hingarajia Jeetkumar B., Enroll:230823030,Class-A,RollNo:11]

[Verified by:Masot Kushal M. Enroll:230823068 Class-A RollNo:32]

Named routes is an important feature in the Laravel framework. It allows you to refer to the routes when generating URLs or redirects to the specific routes. In short, we can say that the naming route is the way of providing a nickname to the route.

- Syntax of defining naming routes:

We can define the named routes by chaining the name method onto the route definition:

```
Route::get('/mypost', function()
{
    //
}) -> name('Home');
```


We can also specify the named routes for controller actions:

```
Route::get('/mypost', [MyController::class,'Home']) -> name('Home');
```

- Generating URLs to named routes

Once you assigned a named route to a given route, then you can use the name of the route while generating URLs or redirecting through a global route function.

```
//Generating URLs
$url= route('Home');
//Generating Redirects...
return redirect() -> route('Home');
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q-12.Explain how to create a controller in laravel.


[Name:Idariya Dhara, Enroll:230823031,Class-A,Roll No:12]

[Verified: Moliya Krushi, Enroll:230823072, Class-A, Roll No:33]

Syntax:

Php artisan make:controller controller_name

Example: Php artisan make:controller Mycontroller

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q-13 Explain Resource Controller in laravel

[Name: Jadav Hetali, Enrollment: 230823032, Class: A, Roll: 13]

Backend developers construct standardized projects on a regular basis as part of our day-to-day work, particularly for CRUD operations. We give each of them their own specific tasks within our controller, and we also build routes for each, which can be a time-consuming and, at times, difficult process.

This procedure requires more of our time and effort, and consequently, it lengthens the amount of time it takes to produce any given product or complete any given job. CRUD operations are very common, and everyone here is familiar with them.

C = Create

R= Retrieve

U= Update

D= Delete

Laravel Controller provides a facility for generating a controller with all these CRUD operations prebuilt with a single line comment, and this controller is known as Resource Controller.

Here is the command for the Resource controller:


`php artisan make: controller controllerName –resource`

Actions of Resource Controller

A resource controller provides us with a total of seven methods with different actions

These actions are:

- index
- create
- store
- show
- edit
- update
- Destroy

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Initializing the resource controller in our routing file:

We can initialize the resource controller routes by using the below snippet in our web.php file, which is located in the routes folder of our Laravel project using the below snippet, i.e.,

use App\Http\Controllers\controllerName;

Route::resource('photos', controllerName::class);

Let's take a look at the seven actions that a resource controller provides to us:

1. Index Method

Route Name: controllerName.index

Request Method: GET

The pages of all the records are displayed using this method. If you have a product page, for example, you will use the index method to retrieve all of the records and show them in the view in the appropriate order.

```
public function index()
{
    //
}
```

2. Create Method

Route Name: controllerName.create

Request Method : GET

This method is used to display the form for creating the record in the table. For example, if you have a product table, we will create a form for the user to use to enter values for the products in the database, and then we will display that form to the user.



```
public function create()
```

```
{
```

```
//
```

```
}
```

3. Store Method

Route Name: controllerName.store

Request Method: POST


This method is used to store the record in the table. For example, if you have a product table, then insert values in the products table by sending the params in a POST request in the route.

```
public function store(Request $request)
```

```
{
```

```
//
```

```
}
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

4. Edit Method

Route Name: controllerName.edit

Request Method: GET

This method is used to show the editing form for a specified resource. This function takes the id as an argument. It can be the primary key or any unique key of the table.

```
/**
 * Show the form for editing the specified resource.
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function edit($id)

{

//

}
```

5. Show Method

Route Name: controllerName.show

Request Method: GET

Used to show the results for the specified resource from the database.



```
public function show($id)
```

```
{
```

```
//
```

```
}
```

6. Update Method

Route Name: controllerName.update

Request Method: PUT/PATCH

Now we use that method to update the specified resource. This method takes arguments by following the PUT or PATCH as a request method followed by a unique id of the record we want to update.



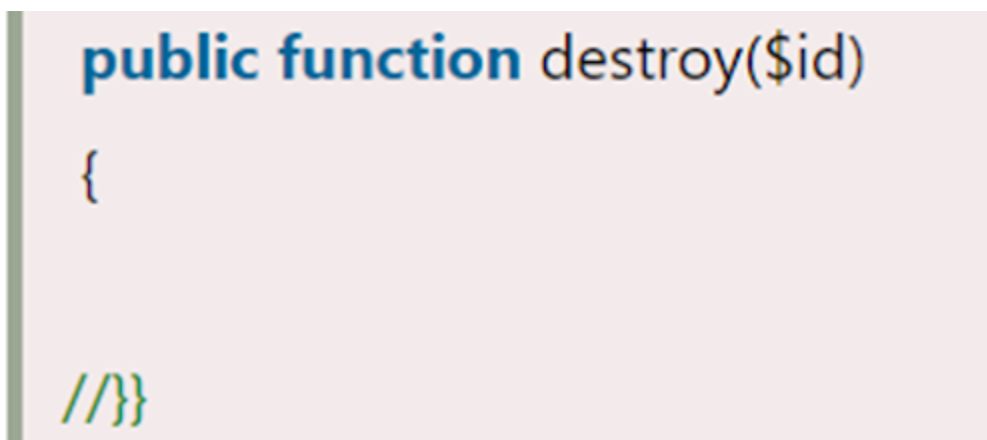
```
/**
 * Update the specified resource in storage.
 * @param \Illuminate\Http\Request $request
 * @param int $id
 * @return \Illuminate\Http\Response
 */
public function update(Request $request, $id)
{
    //
}
```

7. Destroy Method

Route Name: controllerName.delete

Request Method: DELETE

Whenever we create a record, we also want a method and route for deleting this record. By using the destroy method of the resource controller, we can do the same. It takes the id of the table as an identifier for which record you want to delete.



After initializing your resource controller in the routes file, you can check your listed routes by command:

`php artisan route:list`

The outcome in your terminal will be:

```
MINGW64/c:/xampp/htdocs/laravelproject
+-----+-----+-----+-----+-----+
| Domain | Method | URI | Middleware | Name | Action |
+-----+-----+-----+-----+-----+
| | GET|HEAD | api/user | | | Closure |
| | POST | posts | api,auth:api | posts.store | App\Http\Controllers\ |
PostController@store | web | | | | |
| | GET|HEAD | posts | | posts.index | App\Http\Controllers\ |
PostController@index | web | | | | |
| | GET|HEAD | posts/create | | posts.create | App\Http\Controllers\ |
PostController@create | web | | | | |
| | DELETE | posts/{post} | | posts.destroy | App\Http\Controllers\ |
PostController@destroy | web | | | | |
| | PUT|PATCH | posts/{post} | | posts.update | App\Http\Controllers\ |
PostController@update | web | | | | |
| | GET|HEAD | posts/{post} | | posts.show | App\Http\Controllers\ |
PostController@show | web | | | | |
| | GET|HEAD | posts/{post}/edit | | posts.edit | App\Http\Controllers\ |
PostController@edit | web | | | | |
+-----+-----+-----+-----+-----+
admin@SSSIT MINGW64 /c:/xampp/htdocs/laravelproject
```

8. Initializing the Specified Methods

When we initialize the resource controller routes, we can also define only the required routes. For example, if we need only the index and show method, then we can initialize the routes for only these two methods. If our controller name is PhotoController, then the code for doing the same is:



```
use App\Http\Controllers\PhotoController;

Route::resource('photos', PhotoController::class)->only([
    'index', 'show'
]);
```


9. Defining API Resource Routes

We can also define the resource routes of these resource controllers for the APIs. For this, we will follow the below snippet :

```
use App\Http\Controllers\PhotoController;

Route::apiResource('photos', PhotoController::class);
```

So, these are the main aspects of a resource controller.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.14. Explain How to pass data variable from controller to view with example and explanation..

[Name: Vishakha Kasundra, Enrollment: 230823105, Class: A, Roll: 22]

Verified By: [Name: Darshil Nathani, Enrollment: 230823076, Class: A, Roll: 35]

1. Using view():

We can directly pass the data in the ‘view()’ helper function by using the second parameter in the function which takes an array as key and value pair. Note: Comment or delete any previous route in the ‘web.php’ file in ‘routes’ directory

a. Write the below code in the ‘web.php’ file

```
Route::get('/', function () {
    return view('gfg', ['articleName' => 'Article 1']);
});
```

b. Create and write the below code in the ‘gfg.blade.php’ file in ‘resources/views’ directory


```
<!DOCTYPE html>
<html>
<head>
    <title>GeeksforGeeks</title>
</head>
<body>
    <h1>My Article</h1>
    <h2>{{ $articleName }}</h2>
</body>
</html>
```

Here we have used double curly brackets ‘{{ }}’ which is to escape HTML elements. In that, we have specified the name of the key, as a PHP variable, that we passed in the ‘view’ helper function.

2. Using compact():

The ‘compact()’ is a PHP function that can be used for creating an array with variable and their value. Here the name on the variable is the key and the variable content is the value. Note:

Comment or delete any previous route in the ‘web.php’ file in ‘routes’ directory

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

a. Write the below code in the ‘web.php’ file.

```
Route::get('/', function () {
    $articleName = ['Article 1','Article 2'];
    $articlePublished = 'On GeeksforGeeks';
    return view('gfg', compact('articleName',
        'articlePublished'));
});
```

In view(), the first parameter is the name of the view and second is where we have to specify the ‘compact()’ function.


b. Create and write the below code in the ‘gfg.blade.php’ file in ‘resources/views’ directory

```
<!DOCTYPE html>
<html>
<head>
    <title>GeeksforGeeks</title>
</head>
<body>
    <h1>My Article</h1>
    @foreach ($articleName as $article)
        <li>{{ $article }}</li>
    @endforeach
    <p>{{ $articlePublished }}</p>
</body>
</html>
```

Here we have used double curly brackets ‘{{ }}’ which is to escape HTML elements. In that, we have specified the name of the key, as a PHP variable, that we passed in the ‘view’ helper function.

3.Using Controller Class:

Passing data using controller class is easy and is the right way. Note: Comment or delete any previous route in the ‘web.php’ file in ‘routes’ directory.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

a. We first have to create a controller class by running the command below on the command line

```
php artisan make:controller GfGController
```

b. After that, open the ‘GfGController.php’ file in ‘app/Http/Controllers’ directory and create a public function named ‘article’. In this function we can specify any of the data passing method we saw above

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class GfGController extends Controller
{
    public function article() {
        return view('gfg', ['article' =>
            'Passing Data to View']);
    }
}
```


C. Now, write the below code in the ‘web.php’ file.

```
Route::get('/', 'GfGController@article');
```

In this, the first parameter is the route and the second is the controller name with the function name to handle the route which is separated by ‘@’.

d. Create and write the below code in the ‘gfg.blade.php’ file in ‘resources/views’ directory.

```
<!DOCTYPE html>
<html>
<head>
```


	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

```

<title>GeeksforGeeks</title>
</head>
<body>
  <h1>My Article</h1>
  <p>{{ $article }}</p>
</body>
</html>

```

Here we have used double curly brackets ‘{{ }}’ which is to escape HTML elements. In that, we have specified the name of the key, as a PHP variable, that we passed in the ‘view’ helper function.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Que.15 Explain Route Model Binding.

[Name:- Panchani Vishvaben K. ,enrollment no:-230823082, Division:-A ,Roll no:-39]

Verified By: [Name: NAVADIYA RUTVI, Enroll: 230823077, Class: A, Roll: 36]

ANS:

Route Model Binding:-


When injecting a model ID to a route or controller action, you will often query the database to retrieve the model that corresponds to that ID. Laravel route model binding provides a convenient way to automatically inject the model instances directly into your routes. For example, instead of injecting a user's ID, you can inject the entire **User** model instance that matches the given ID.

Implicit Binding:-

Laravel automatically resolves Eloquent models defined in routes or controller actions whose type-hinted variable names match a route segment name. For example:

```
use App\Models\User;

Route::get('/users/{user}', function (User $user) {
    return $user->email;
});
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023


Since the `$user` variable is type-hinted as the `App\Models\User` Eloquent model and the variable name matches the `{user}` URI segment, Laravel will automatically inject the model instance that has an ID matching the corresponding value from the request URI. If a matching model instance is not found in the database, a 404 HTTP response will automatically be generated.

Of course, implicit binding is also possible when using controller methods. Again, note the `{user}` URI segment matches the `$user` variable in the controller which contains an `App\Models\User` type-hint:

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

// Route definition...
Route::get('/users/{user}', [UserController::class, 'show']);


// Controller method definition...
public function show(User $user)
{
    return view('user.profile', ['user' => $user]);
}
```


	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Que.16 Explain csrf token in laravel

Verified By: [Name: Palwar Mahaveersinh, Enroll: 230823080, Class: A, Roll: 37]

Ans: This token is used to verify that the authenticated user is the person actually making the requests to the application.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023


Q-17 Explain Request Headers OR explain method facking in laravel.

[Name:-Kaila Mansi ,enrollment no:-230823041, Division:-A ,Roll no:-17]

Request Headers : Request headers are additional information that are sent along with an HTTP request. They can be used to specify the content type, encoding, authentication, and other aspects of the request .

```
$userAgent = $request->header('User-Agent');
```

Method facking : Method facking in Laravel is a technique to mock or spy on certain aspects of your application, such as facades, events, or jobs, so they are not actually executed during a test. This allows you to isolate the code under test and verify its behavior without side effects.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q.18. Explain responses in view.

[Name: Parth Kakdiya, Enrollment: 230823042, Class: A, Roll: 18]

Verified By: [Name: Vishva Panchani, Enrollment: 230823082, Class: A, Roll: 39]


ANS.

Introduction:

In Laravel, an HTML response can be returned by using the `response()` helper function and calling the `view()` function onto it. The `view()` function takes the name of the view as its first argument and an array of data as its second argument.

Views separate your controller / application logic from your presentation logic and are stored in the `resources/views` directory. When using Laravel, view templates are usually written using the Blade templating language. A simple view might look something like this:

```
<html>
  <body>
    <h1>Hello, {{ $name }}</h1>
  </body>
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

</html>

Since this view is stored at resources/views/greeting.blade.php, we may return it using the global view helper like so:

```
Route::get('/', function () {
    return view('greeting', ['name' => 'James']);
});
```

Strings and Arrays

All routes and controllers should return a response to be sent back to the user's browser. Laravel provides several different ways to return responses. The most basic response is returning a string from a route or controller. The framework will automatically convert the string into a full HTTP response:

```
Route::get('/', function () {
    return 'Hello World';
});
```

In addition to returning strings from your routes and controllers, you may also return arrays. The framework will automatically convert the array into a JSON response:

```
Route::get('/', function () {
    return [1, 2, 3];
});
```

Redirects

Redirect responses are instances of the `Illuminate\Http\RedirectResponse` class, and contain the proper headers needed to redirect the user to another URL. There are several ways to generate a `RedirectResponse` instance. The simplest method is to use the global redirect helper:


```
Route::get('/dashboard', function () {
    return redirect('home/dashboard');
});
```

Redirecting to Named Routes

When you call the redirect helper with no parameters, an instance of `Illuminate\Routing\Redirector` is returned, allowing you to call any method on the `Redirector` instance. For example, to generate a `RedirectResponse` to a named route, you may use the `route` method:

```
return redirect()->route('login');
```

If your route has parameters, you may pass them as the second argument to the `route` method:


	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

```
return redirect()->route('profile', ['id' => 1]);
```

View Responses

If you need control over the response's status and headers but also need to return a view as the response's content, you should use the view method:

```
return response()
->view('hello', $data, 200)
->header('Content-Type', $type);
```

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Q-19. @yield, @section, @include, @Stack, @push, @if, @if-else @empty, @foreach, @forelse, @while, @switch (blade_directories)?

[Name: kalariya vishva, Enrollment: 230823044, Class: A, Roll: 19]

Ans:-

@yield

The Yield option in laravel is used to define the section in a particular layout and is consistently used to retrieve content from the child page to a master page. So if the Laravel executes a blade file, it just verifies if the user has the extended layout and then pushes the master layout which commences from getting the @section.

Example:

```
<body> @yield ('content')
</body>
```

@section

If the user wants to yield a directive in the master template, he can imply @section in the partial pages. The master template fetches dynamic data from the partial and displays it in the browser section. In the partial section, use the below directive,

Example:

```
@section ('content')
```

@include


@include is just like a basic PHP include, it includes a "partial" view into your view.

@extends lets you "extend" a template, which defines its own sections etc. A template that you can extend will define its own sections using @yield, which you can then put your own stuff into in your view file.

Example:

```
@include('layouts.masters.main')
@extends('layouts.masters.main')
```

@push

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

The **@push** and **@stack** commands have been in Laravel for a year now, but have not made it to any documentation yet. I only discovered them by accident.

Example:

```
@push('script-head')
<script>
    ...do some stuff here that this view needs...
</script>
@endpush
```

If Statements

You may construct **if** statements using the **@if**, **@elseif**, **@else**, and **@endif** directives. These directives function identically to their PHP counterparts:

Example:

```
@if (count($records) === 1)
    I have one record!
@endif
@elseif (count($records) > 1)
    I have multiple records!
@else
    I don't have any records!
@endif
```

Loops


In addition to conditional statements, Blade provides simple directives for working with PHP's supported loop structures. Again, each of these directives functions identically to their PHP counterparts:

Example:

```
@for ($i = 0; $i < 10; $i++)
    The current value is {{ $i }}
@endfor
@foreach ($users as $user)
    <p>This is user {{ $user->id }}</p>
@endforeach
```

Switch

Switch statements can be constructed using the **@switch**, **@case**, **@break**, **@default** and **@endswitch** directives:

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Example:

@switch(\$i)

@case(1)

First case...

@break

@case(2)


Second case...

@break

@default

Default case...

@endswitch

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023


Q.20. Explain json responses in laravel.

[Name: Kanjariya Rahul M, Enroll: 230823045, Class: A, Roll: 20]

Verified by[name:parmar bhoomi Enroll:230823085, Class:a, Roll:41]

Ans : JSON response in Laravel involves returning data from a route or controller in JSON format, typically achieved by using Laravel's built-in `response()->json()` method.

Alternatively, you can also return JSON responses directly from routes using Laravel's `Route::get()` or `Route::post()` methods.

	ATMIYA University Faculty of Science Department of Computer Application & Department of Computer Science	
	23PGCA203-Core-Laravel For Modern Web Development	Rev. No: 001/Dec-2023

Question 21 **What are blade templates? Explain blade templates in laravel.**

[Name:-Yashavi kariya ,enrollment no:-230823049, Division:-A ,Roll no:-21]

Verified By: [Name: Paresh Parmar, Enroll: 230823086, Class: A, Roll: 42]

The Blade is a powerful templating engine in a Laravel framework. The blade allows the templating engine to be used easily, and it makes the syntax writing very simple.

The blade templating engine provides its own structure such as conditional statements and loops. To create a blade template, you just need to create a view file and save it with a .blade.php extension instead of .php extension. The blade templates are stored in the /resources/view directory.

The main advantage of using the blade template is that we can create the master template, which can be extended by other files.

Displaying data

you want to print the value of a variable, then you can do so by simply enclosing the variable within the curly brackets.

Syntax

If `{{ $variable }}`;

In the blade template, we do not need to write the code between `<?php echo $variable; ?>`. The above syntax is equivalent to `<?= $variable ?>`.