

Name: Md. Ruhul Kuddus

Assignment: Module-14

Question-1:

You have a Laravel application with a form that submits user information using a POST request. Write the code to retrieve the 'name' input field value from the request and store it in a variable called \$name.

Route Code:

```
Route::post( uri: '/userinfo', [AssignmentController::class, 'UserInfo'] );
```

Controller Code:

```
function UserInfo(Request $request):string
{
    $name = $request->input( key: 'name' );
    return $name;
}
```

Postman output:

The screenshot shows the Postman interface with a new collection named 'Question-1'. A POST request is configured with the URL 'http://127.0.0.1:8000/api/userinfo?name=Ruhul Kuddus'. The 'Query Params' tab is active, showing a single parameter 'name' with the value 'Ruhul Kuddus'. The 'Body' tab is also active, showing the response 'Ruhul Kuddus'. The status is '200 OK'.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> name	Ruhul Kuddus	
Key	Value	Description

Body: 1 Ruhul Kuddus

Question-2:

In your Laravel application, you want to retrieve the value of the 'User-Agent' header from the current request. Write the code to accomplish this and store the value in a variable called \$userAgent.

Route:

```
Route::post('uri: '/useragent',[AssignmentController::class,'UserAgent']);
```

Controller:

```
function UserAgent(Request $request):string
{
    $userAgent = $request->header('User-Agent');
    return $userAgent;
}
```

Postman:

The screenshot shows the Postman interface with a collection named 'New Collection'. The selected request is 'GET Question-2-(user-agent)'. The URL is 'http://127.0.0.1:8000/api/useragent'. The 'Headers' tab is active, displaying 8 headers:

Key	Value	Description
Content-Length	0	
Host	<calculated when request is sent>	
User-Agent	PostmanRuntime/7.31.1	
Accept	*/*	
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	

The 'Body' tab is also visible, showing the response body: 'PostmanRuntime/7.31.1'. The status is '200 OK', time is '143 ms', and size is '3'.

Question-3:

You are building an API endpoint in Laravel that accepts a GET request with a 'page' query parameter. Write the code to retrieve the value of the 'page' parameter from the current request and store it in a variable called \$page. If the parameter is not present, set \$page to null.

Route:

```
Route::get(uri: '/query',[AssignmentController::class,'ApiEndpoint']);
```

Controller:

```
function ApiEndpoint(Request $request)
{
    $page = $request->query( key: 'page', default: null);
    if($page !== null){
        return $page;
    }else{
        return;
    }
}
```

Postman:

The screenshot shows the Postman interface with a GET request configured. The URL is `http://127.0.0.1:8000/api/query?page=newPage`. Under the 'Query Params' tab, a parameter 'page' is added with the value 'newPage'. The response status is '200 OK' and the body is 'newPage'.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> page	newPage	
Key	Value	Description

Body: newPage

Question-4:

Create a JSON response in Laravel with the following data:

```
{  
  "message": "Success",  
  "data": { "name": "John Doe", "age": 25 }  
}
```

Route:

```
Route::get( uri: '/jsonresponse', [AssignmentController::class, 'JsonResponse'] );
```

Controller:

```
function JsonResponse():JsonResponse  
{  
    $data = array(  
        "message"=> "Success",  
        "data"=>array(  
            "name"=> "John Doe",  
            "age"=> 25  
        )  
    );  
    return response()->json($data);  
}
```

Postman:

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/jsonresponse`. The request is successful, returning a 200 OK status with a response time of 385 ms and a size of 364 B. The response body is displayed in JSON format, showing the following structure:

```
{  
  "message": "Success",  
  "data": {  
    "name": "John Doe",  
    "age": 25  
  }  
}
```

Question-5:

You are implementing a file upload feature in your Laravel application. Write the code to handle a file upload named 'avatar' in the current request and store the uploaded file in the 'public/uploads' directory. Use the original filename for the uploaded file.

Route:

```
Route::post(uri: '/upload',[AssignmentController::class,'FileUploads']);
```

Controller:

```
function FileUploads(Request $request):bool
{
    $file=$request->file(key: 'avatar');
    $file->move(public_path(path: 'uploads'),$file->getClientOriginalName());
    return true;
}
```

Postman:

GET	http://127.0.0.1:8000/api/upload	Send
Params	Authorization	Headers (8)
Body	Pre-request Script	Tests
Settings		Cookies
none	form-data	x-www-form-urlencoded
raw	binary	GraphQL
KEY	VALUE	DESCRIPTION
avatar	330931843.jpg	
Key	Value	Description

Question 6:

Retrieve the value of the 'remember_token' cookie from the current request in Laravel and store it in a variable called \$rememberToken. If the cookie is not present, set \$rememberToken to null.

Route:

```
Route::post('uri: '/setcookie', [AssignmentController::class, 'SetCookie']);
```

Controller:

```
function SetCookie(Request $request)
{
    $rememberToken = $request->cookie(key: 'remember_token', default: null);
    return $rememberToken;
}
```

Postman:

GET ⌵ http://127.0.0.1:8000/api/setcookie Send ⌵

Params Authorization Headers (8) Body ● Pre-request Script Tests Settings Cookies

Query Params

	KEY	VALUE	DESCRIPTION	⋮	Bulk Edit
	Key	Value	Description		

Question 7:

Create a route in Laravel that handles a POST request to the '/submit' URL. Inside the route closure, retrieve the 'email' input parameter from the request and store it in a variable called \$email.

Return a JSON response with the following data: { "success": true, "message": "Form submitted successfully." }

Route:

```
Route::post('uri: '/submit',function(Request $request){
    $email = $request->input( key: 'email');
    if($email){
        return array(
            "success"=> true,
            "message"=> "Form submitted successfully."
        );
    }else{
        return "Email must not be empty!";
    }
});
```

Postman:

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/submit`. The 'Body' tab is selected, and the 'form-data' radio button is chosen. A table below lists the form data parameters.

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> email	ruhul@gmail.com	
Key	Value	Description