

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.

Web.php

```
//Question 1:  
Route::post('/retriveName', [AssignmentController::class, 'retriveName']);
```

Corresponding controller Method:

```
function retriveName(Request $request)  
{  
    $name = $request->input('name');  
    return $name;  
}
```

Postman Output:

The screenshot shows the Postman interface with a POST request to `{{baseurl}}/retriveName` using the `form-data` type. The request body contains a single parameter: `name` with the value `sazzad`. The response status is `200 OK` with a time of `544 ms` and a size of `394 B`. The response body is displayed in the `Pretty` format as `1 sazzad`.

Key	Value	Description
<input checked="" type="checkbox"/> name	sazzad	

Body: 1 sazzad

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.

```
//Question 2:  
Route::post('/userAgent', [AssignmentController::class, 'userAgent']);
```

Corresponding controller Method:

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

Postman Output:

The screenshot shows the Postman interface with a POST request configured to retrieve the User-Agent header. The request is sent to the endpoint `{{baseurl}}/userAgent`. The Headers tab is active, showing the following headers:

Key	Value	Description
Accept-Encoding	gzip, deflate, br	
Connection	keep-alive	
User-Agent	Sazzad	

The response is displayed in the Body tab, showing the value `Sazzad`. The status is 200 OK, the time is 363 ms, and the size is 394 B.

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.

```
//Question 3:  
Route::get('/queryParam', [AssignmentController::class, 'page']);
```

Corresponding controller Method:

```
public function page( Request $request){  
    $page = $request->query('page', null);  
    if($page){  
        return $page;  
    }else{  
        return "please set the page number";  
    }  
}
```

Postman output:

The screenshot shows the Postman interface with a GET request to `{{baseurl}}/queryParam?page=sazzadur`. The request is successful with a status of 200 OK. The response body is `sazzadur`.

Key	Value	Description
page	sazzadur	

Body: `sazzadur`

Question 4:

Create a JSON response in Laravel with the following data:

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

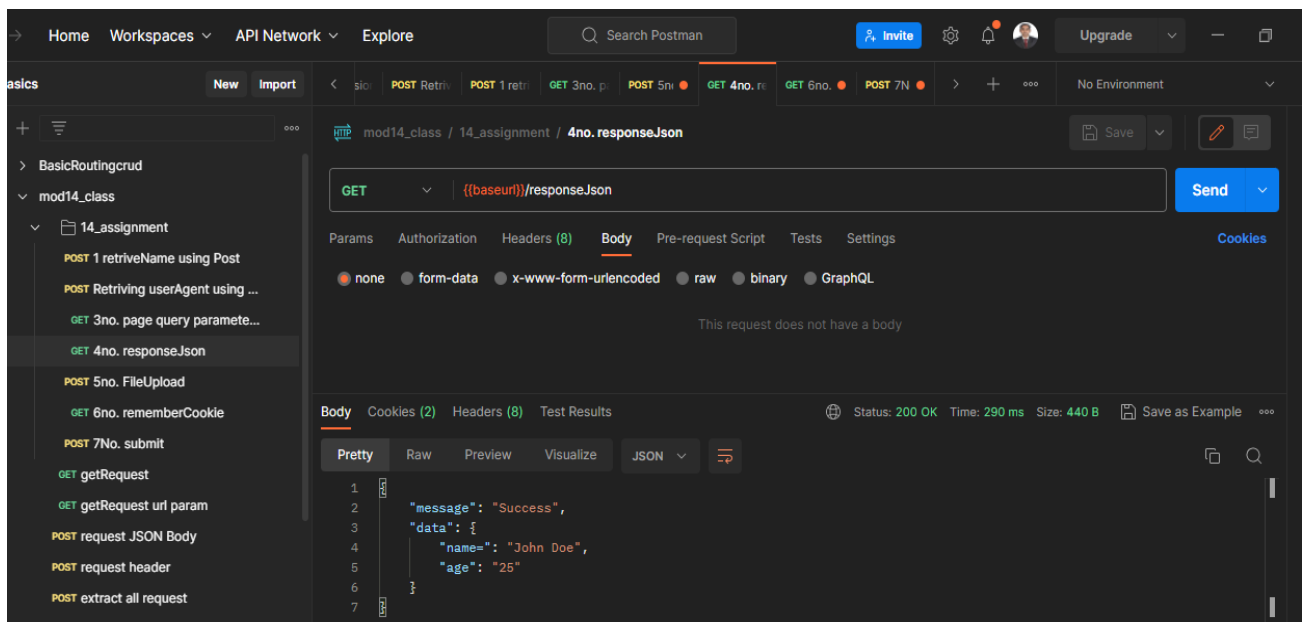
Web.php

```
//Question 4:
Route::get('/responseJson', [AssignmentController::class, 'responseJson']);
```

Corresponding controller Method:

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

Postman output:



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.

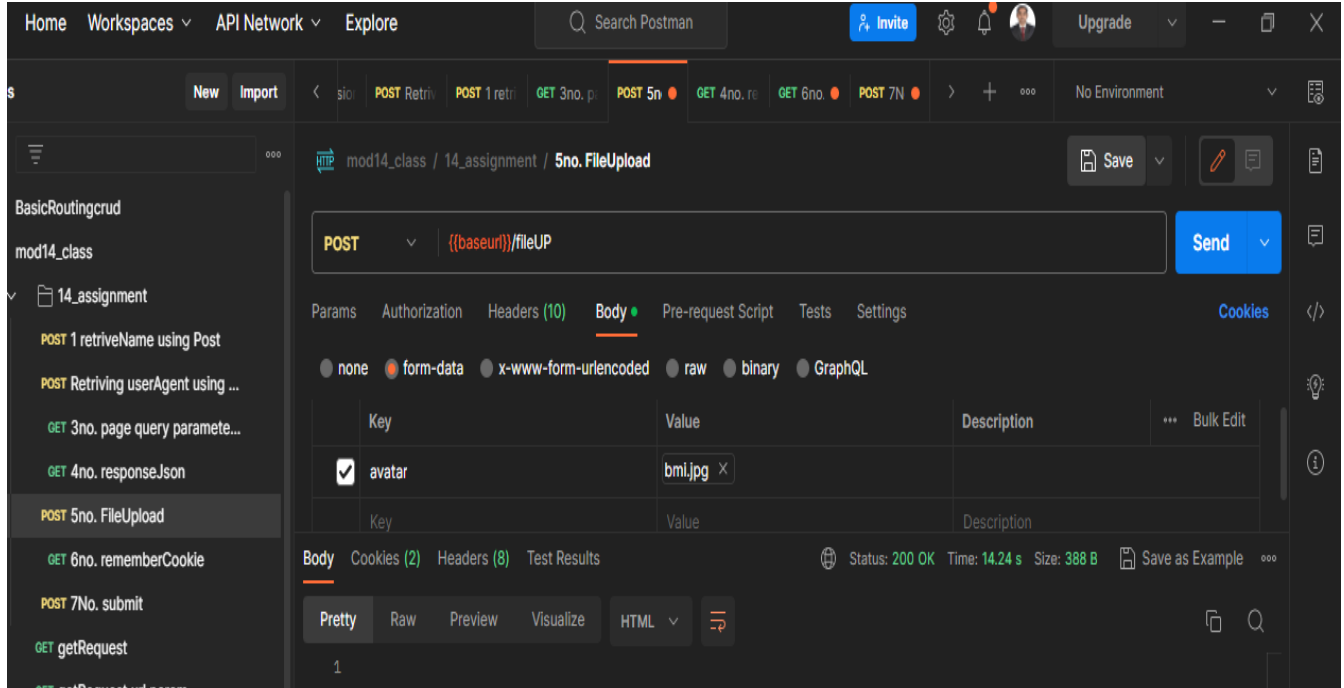
Web.php

```
//Question 5:  
Route::post('/fileUP', [AssignmentController::class, 'fileUP']);
```

Corresponding controller Method:

```
function fileUP(Request $request)  
{  
    $img = $request->file('avatar');  
    $photoFile = $img->move(public_path('uploads'), $img->getClientOriginalName());  
}
```

Postman output:



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.

Web.php

```
//Question 6:  
Route::get('/rememberCookie', [AssignmentController::class, 'rememberCookie']);
```

Corresponding controller Method:

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

Postman output:

The screenshot shows the Postman interface with a GET request to `{{baseurl}}/rememberCookie` executed successfully. The status is 200 OK, with a response time of 354 ms and a body size of 416 B. The response body, displayed in 'Pretty' format, contains the text '1 hello from remembered cookie'.

Query Params Table:

Key	Value	Description
Key	Value	Description

Body: 1 hello from remembered cookie

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."
}
```

web.php

```
//Question 7:
Route::post('/submit', function (Request $request) {
    $email = $request->input('email');
    if($email){
        $responseData = [
            'success' => true,
            'message' => 'Form submitted successfully.'
        ];
        return response()->json($responseData);
    }else{
        return "Please Enter Email";
    }
});
```

Postman output:

The screenshot shows the Postman interface with a POST request to `{{baseurl}}/submit` using form-data. The 'email' parameter is set to `zitu094@gmail.com`. The response is a 200 OK status with a JSON body: `{ "success": true, "message": "Form submitted successfully." }`.

Key	Value	Description
email	zitu094@gmail.com	

Body: `{ "success": true, "message": "Form submitted successfully." }`