

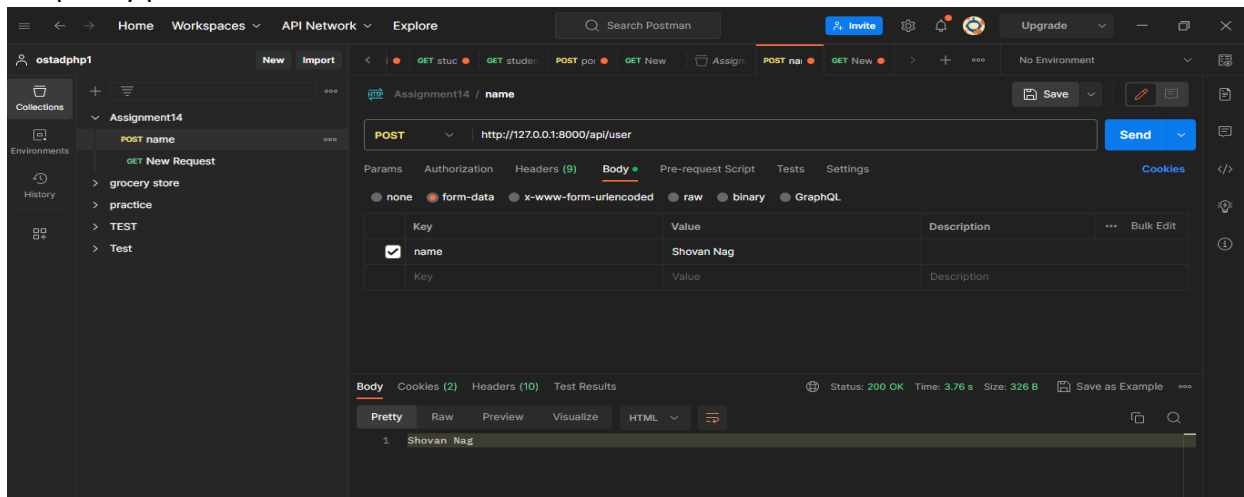
Assignment-14

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.

```
// Question 1
1 reference | 0 overrides
function UserName(Request $request):string{
    $name = $request->input('name');
    return $name;
}
```

```
Route::post('/user',[AppController::class, 'UserName']);
```

Output by postman:

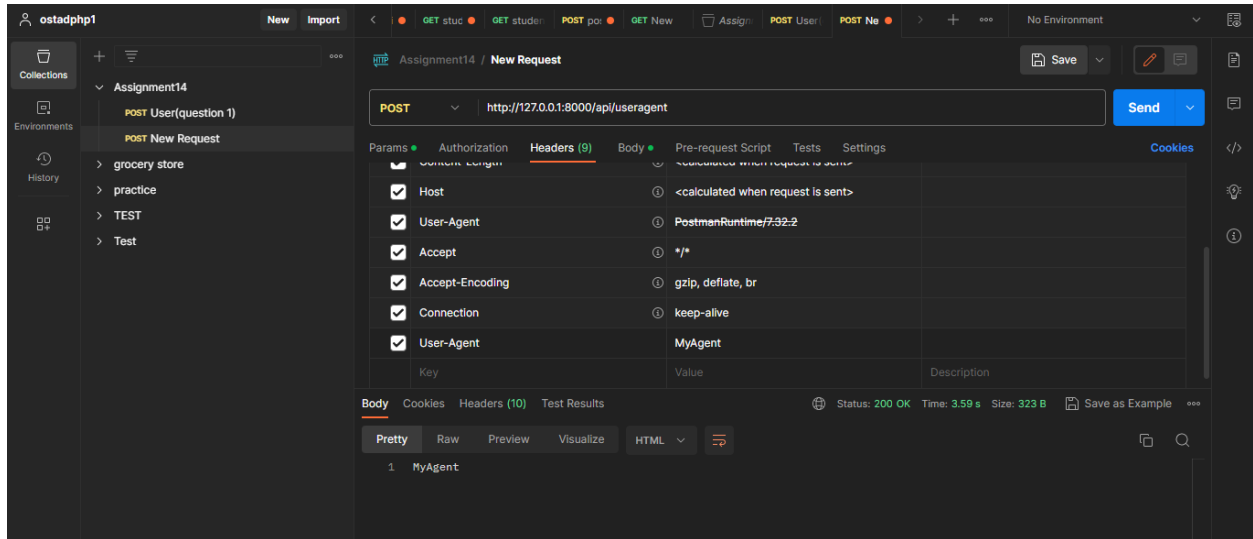


Question 2: In your Laravel application, you want to retrieve the value of the 'User-Agent' header from the current request. Write the code to accomplish this and store the value in a variable called \$userAgent

```
// Question 2
1 reference | 0 overrides
function userAgent(Request $request):string{
    $userAgent = $request->header('User-Agent');
    return $userAgent;
}
```

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

Output by postman:

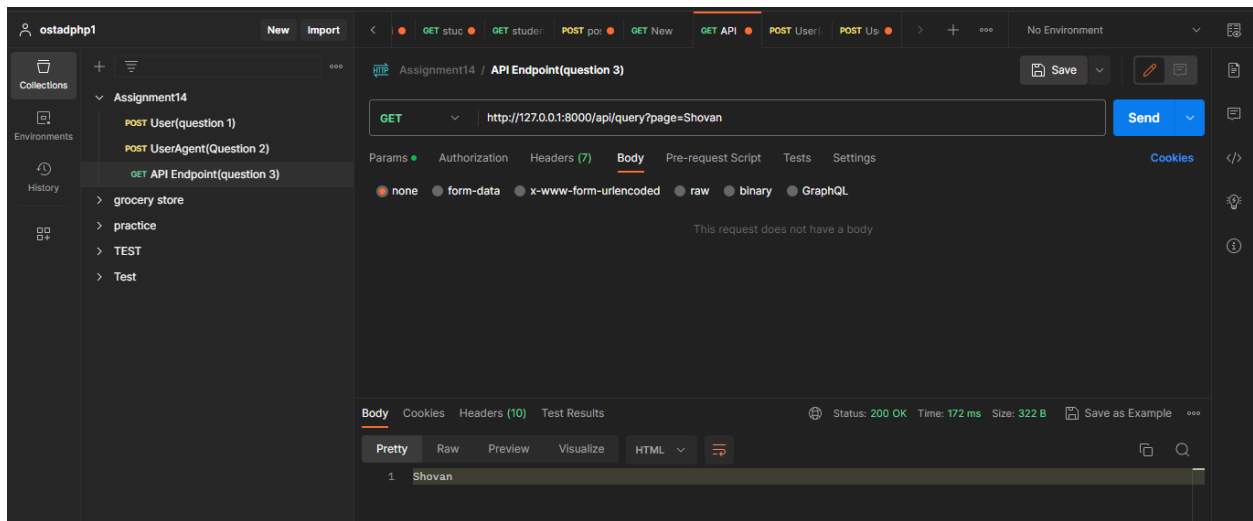


Question 3: You are building an API endpoint in Laravel that accepts a GET request with a 'page' query parameter. Write the code to retrieve the value of the 'page' parameter from the current request and store it in a variable called \$page. If the parameter is not present, set \$page to null.

```
// Question 3
1 reference | 0 overrides
function ApiEndpoint(Request $request){
    $page = $request->query('page',null);
    if($page !== null){
        return $page;
    }else{
        return;
    }
}
```

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

Output by postman:



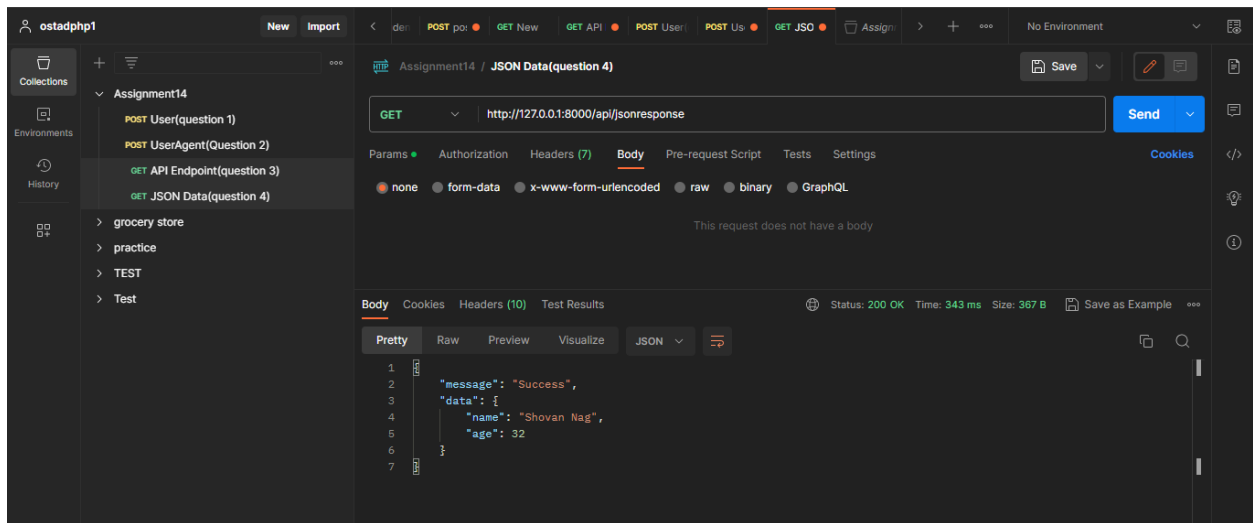
Question 4: Create a JSON response in Laravel with the following data:

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

```
// Question 4
1 reference | 0 overrides
function JsonResponse(Request $request):jsonresponse{
    $data = array(
        "message"=> "Success",
        "data"=>array(
            "name"=> "Shovan Nag",
            "age"=> 32
        )
    );
    return response()->json($data);
}
```

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

Output by postman:

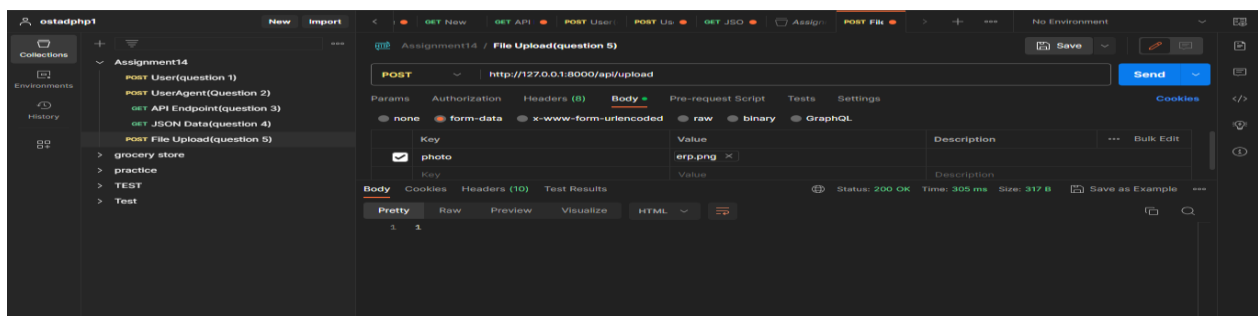


Question 5: You are implementing a file upload feature in your Laravel application. Write the code to handle a file upload named 'avatar' in the current request and store the uploaded file in the 'public/uploads' directory. Use the original filename for the uploaded file.

```
// Question 5
1 reference | 0 overrides
function FileUploads(Request $request):bool{
    $file=$request->file('photo');
    $file->move(public_path('uploads'),$file->getClientOriginalName());
    return true;
}

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

Output by postman:



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.

```
// Question 6
1 reference | 0 overrides
function SetCookie(Request $request){
    $rememberToken = $request->cookie('remember_token',null);
    return $rememberToken;
}
```

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

Output by postman:

The screenshot shows the Postman interface with a POST request to `http://127.0.0.1:8000/ap/setcookie`. The request headers include `Accept-Encoding` and `Connection: keep-alive`. The response status is `200 OK` with a time of `8.01 s` and size of `316 B`. The `Cookies` tab shows a single cookie:

Name	Value	Domain	Path	Expires	HttpOnly	Secure
myCookie	value	127.0.0.1	/	Tue, 21 May 2...	false	false

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."  
}
```

```
// Question 7  
Route::post('/submit',function(Request $request){  
    $email = $request->input('email');  
    if($email){  
        return array(  
            "success"=> true,  
            "message"=> "Form submitted successfully."  
        );  
    }else{  
        return "Email must not be empty!";  
    }  
}
```

Output by postman:

The screenshot shows the Postman interface with a collection named 'Assignment14'. The selected request is 'submit(question 7)' which is a POST request to 'http://127.0.0.1:8000/ap/submit'. The request body is set to 'form-data' with a single parameter 'email' having the value 'shovanr06@gmail.com'. The response is shown in the 'Body' tab, displaying a JSON object:

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

 The status is 200 OK, time is 3.25 s, and size is 365 B.

Key	Value	Description
email	shovanr06@gmail.com	

Key	Value	Description

```
1  {  
2    "success": true,  
3    "message": "Form submitted successfully."  
4  }
```