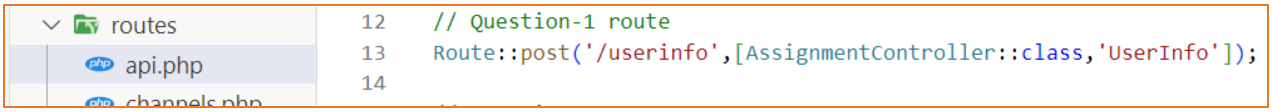
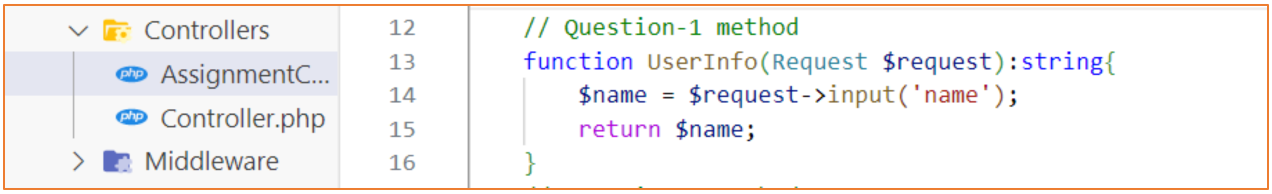
Module 14 Assignment

Syed Amir Hamza

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

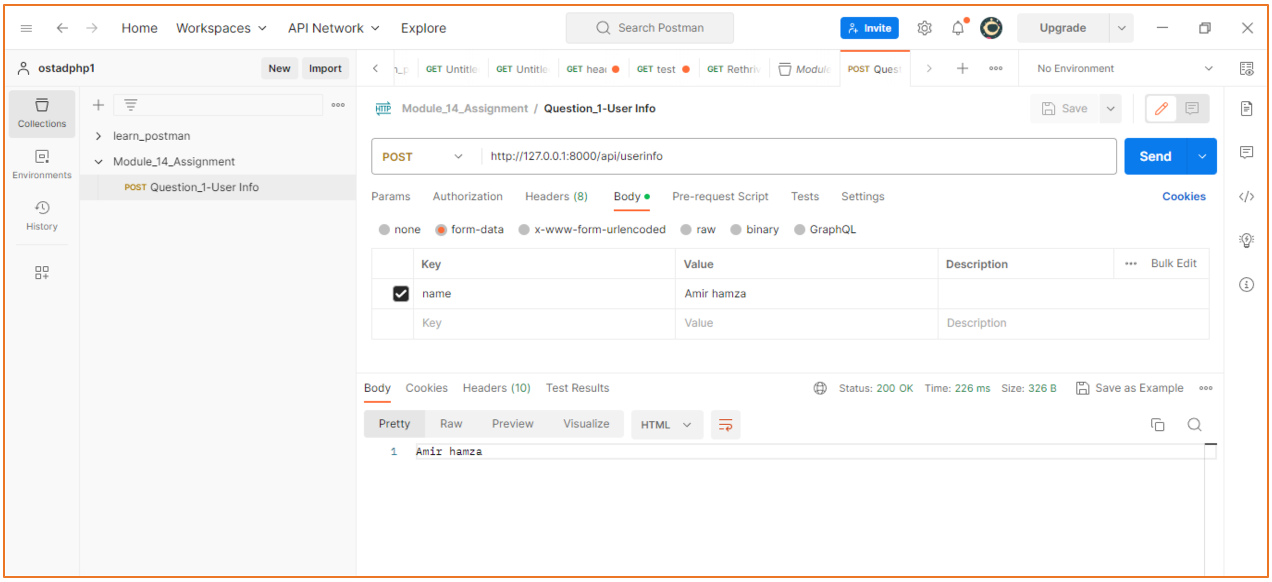


```
// Question-1 route
Route::post('/userinfo',[AssignmentController::class,'UserInfo']);
```



```
// Question-1 method
function UserInfo(Request $request):string{
    $name = $request->input('name');
    return $name;
}
```

Output(postman):



Module_14_Assignment / Question_1-User Info

POST http://127.0.0.1:8000/api/userinfo

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

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

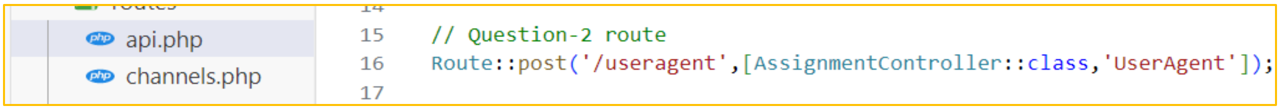
	Key	Value	Description
☑	name	Amir hamza	
	Key	Value	Description

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 226 ms Size: 326 B

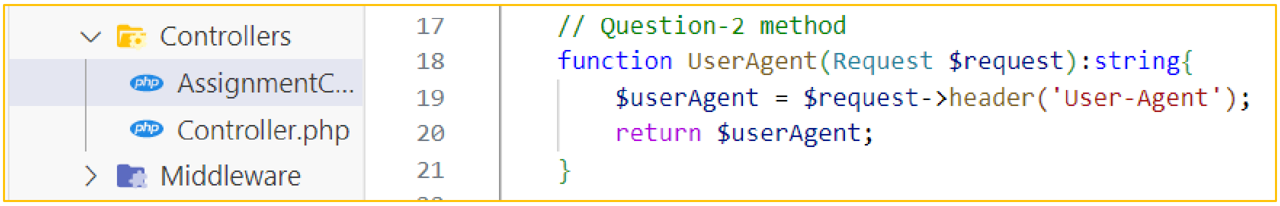
Pretty Raw Preview Visualize HTML

```
Amir hamza
```

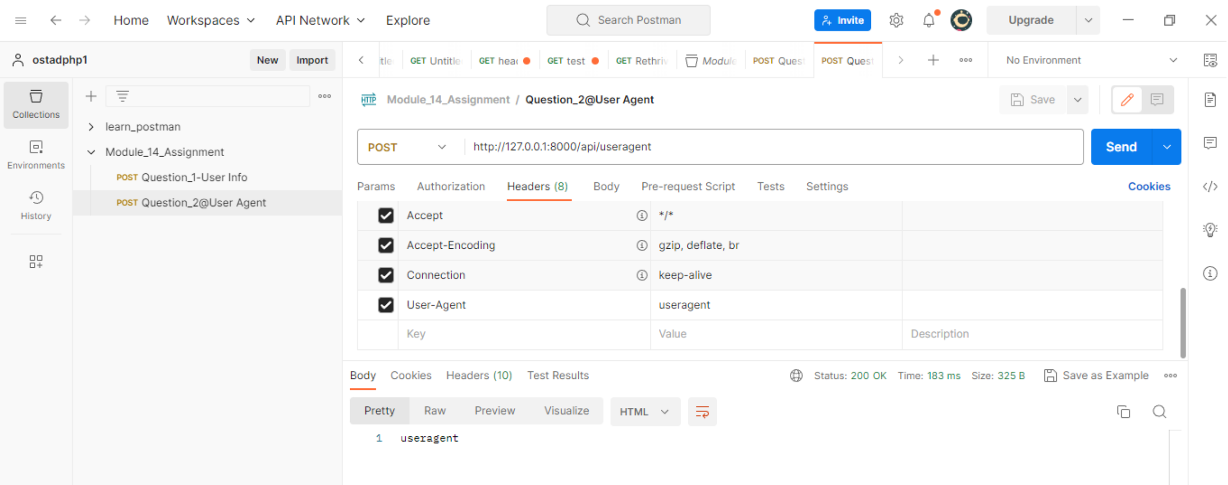
Question 2: In your Laravel application, you want to retrieve the value of the 'User-Agent' header from the current request. Write the code to accomplish this and store the value in a variable called $userAgent.



```
// Question-2 route
Route::post('/useragent',[AssignmentController::class,'UserAgent']);
```



```
// Question-2 method
function UserAgent(Request $request):string{
    $userAgent = $request->header('User-Agent');
    return $userAgent;
}
```



Module_14_Assignment / Question_2@User Agent

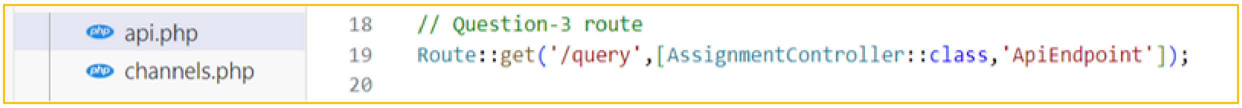
POST http://127.0.0.1:8000/api/useragent Send

	Key	Value	Description
☑	Accept	*/*	
☑	Accept-Encoding	gzip, deflate, br	
☑	Connection	keep-alive	
☑	User-Agent	useragent	

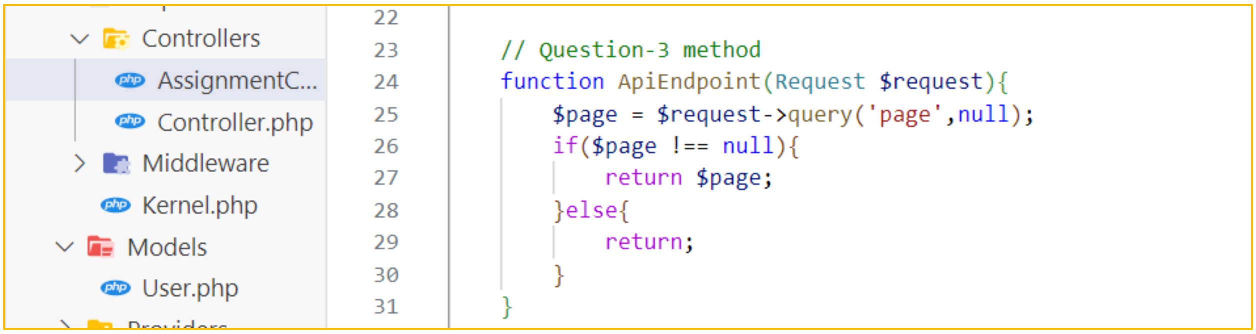
Status: 200 OK Time: 183 ms Size: 325 B

```
1  useragent
```

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.



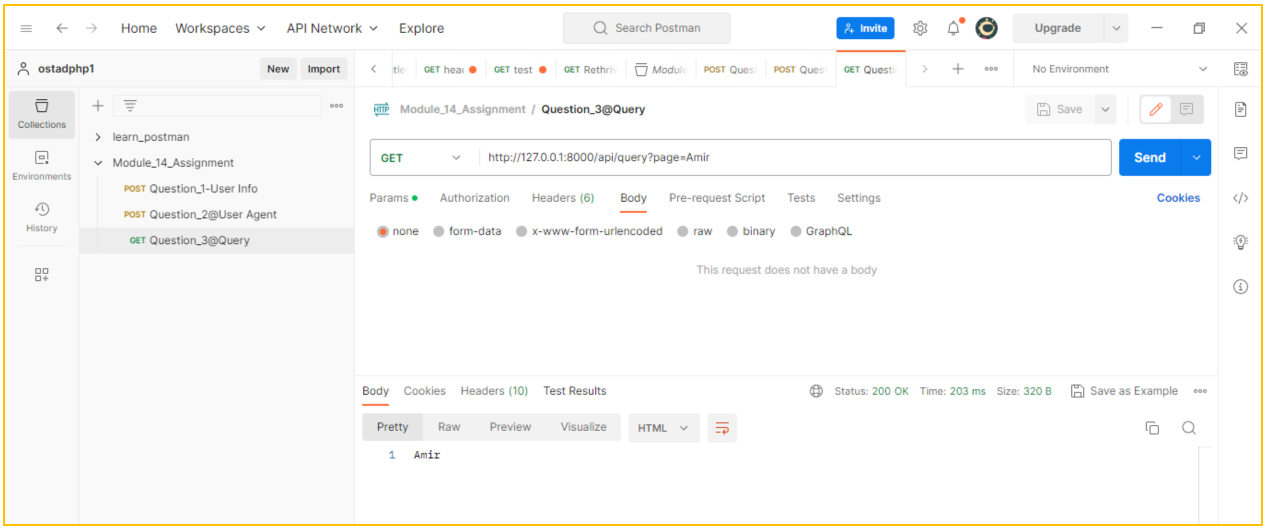
```php
// Question-3 route
Route::get('/query',[AssignmentController::class,'ApiEndpoint']);
```



```php

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

Output(postman):



Module_14_Assignment / Question_3@Query

GET http://127.0.0.1:8000/api/query?page=Amir Send

● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

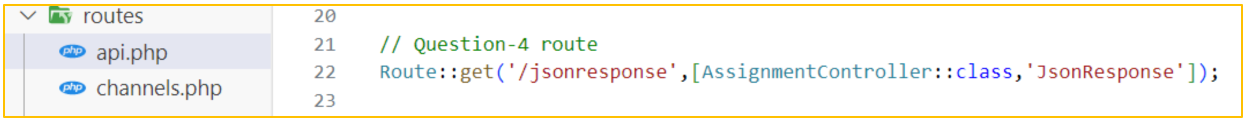
This request does not have a body

Status: 200 OK Time: 203 ms Size: 320 B

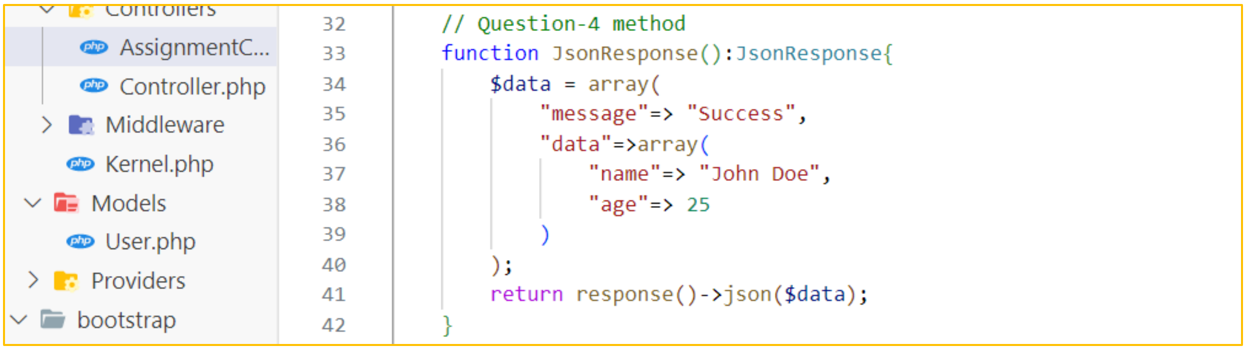
```
1  Amir
```

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

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

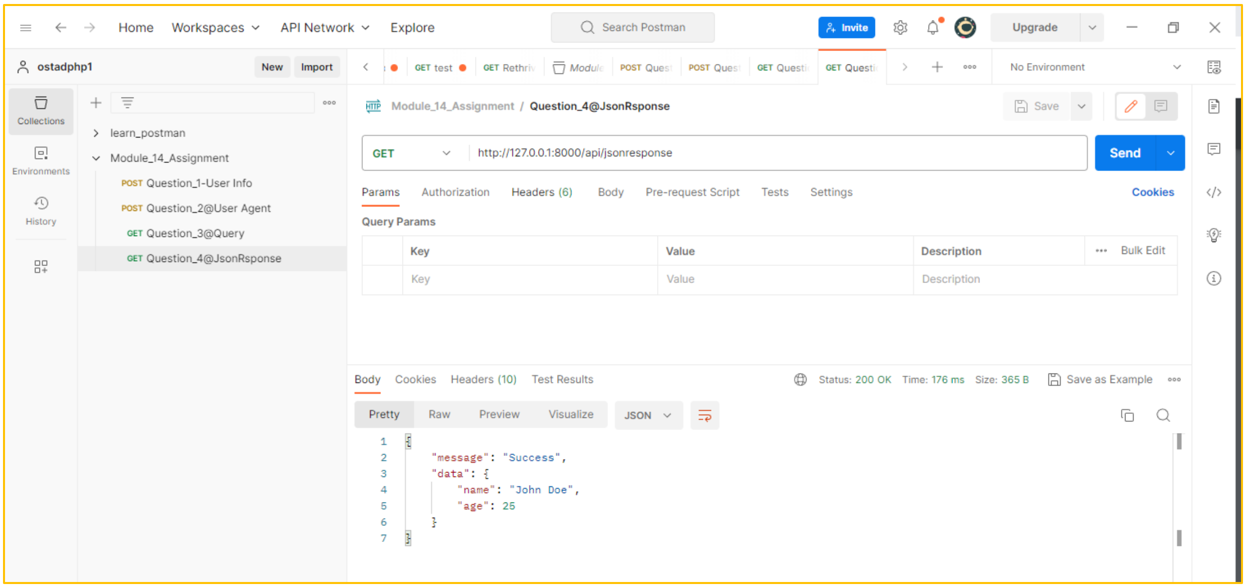


```
// Question-4 route
Route::get('/jsonresponse',[AssignmentController::class,'JsonResponse']);
```



```
    // Question-4 method
    function JsonResponse():JsonResponse{
        $data = array(
            "message"=> "Success",
            "data"=>array(
                "name"=> "John Doe",
                "age"=> 25
            )
        );
        return response()->json($data);
    }
```

Output(postman):

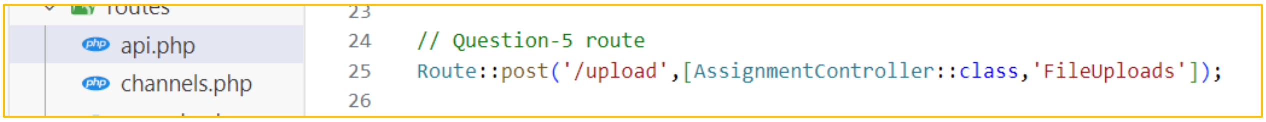


GET http://127.0.0.1:8000/api/jsonresponse

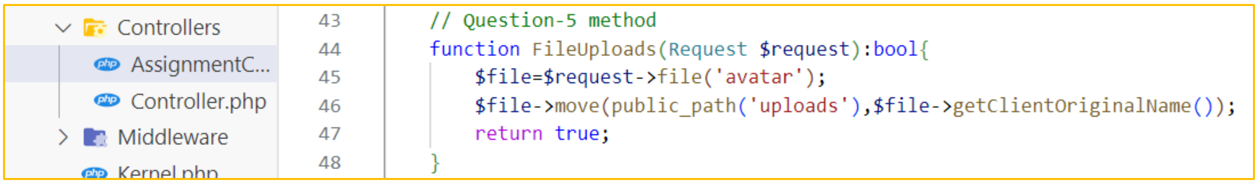
Status: 200 OK Time: 176 ms Size: 365 B

```
{
    "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

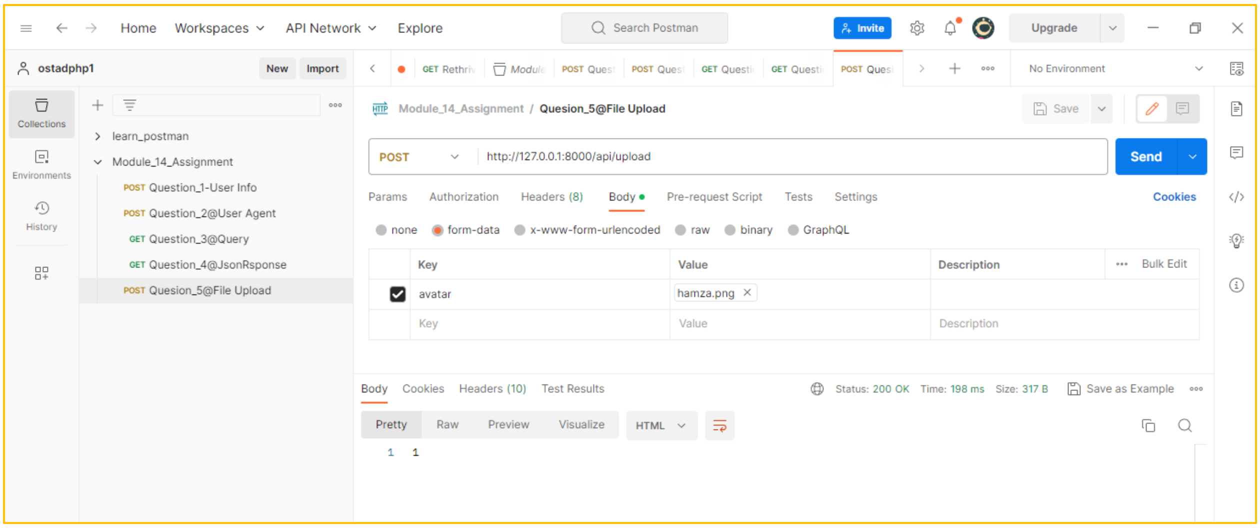


```
// Question-5 route
Route::post('/upload',[AssignmentController::class,'FileUploads']);
```



```
// Question-5 method
function FileUploads(Request $request):bool{
    $file=$request->file('avatar');
    $file->move(public_path('uploads'),$file->getClientOriginalName());
    return true;
}
```

Output(postman):



Module_14_Assignment / Quesion_5@File Upload

POST http://127.0.0.1:8000/api/upload

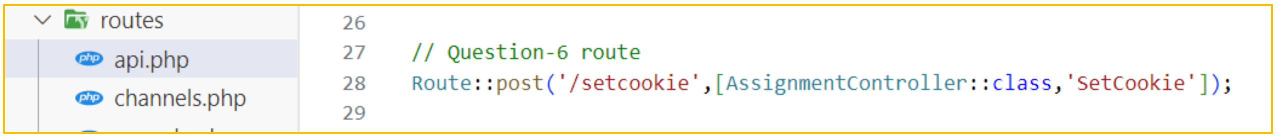
Body: ○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key	Value	Description
☑	avatar	hamza.png	
	Key	Value	Description

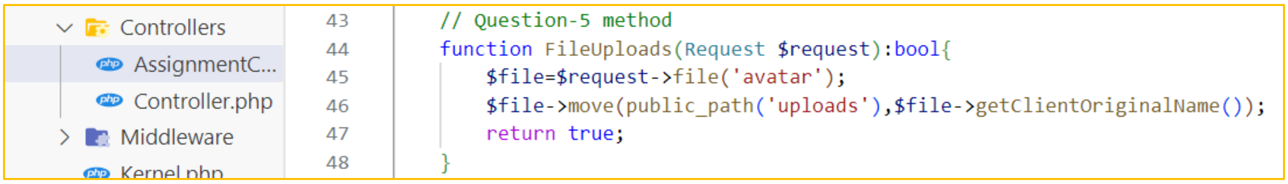
Body Cookies Headers (10) Test Results — Status: 200 OK Time: 198 ms Size: 317 B

```
1
```

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.

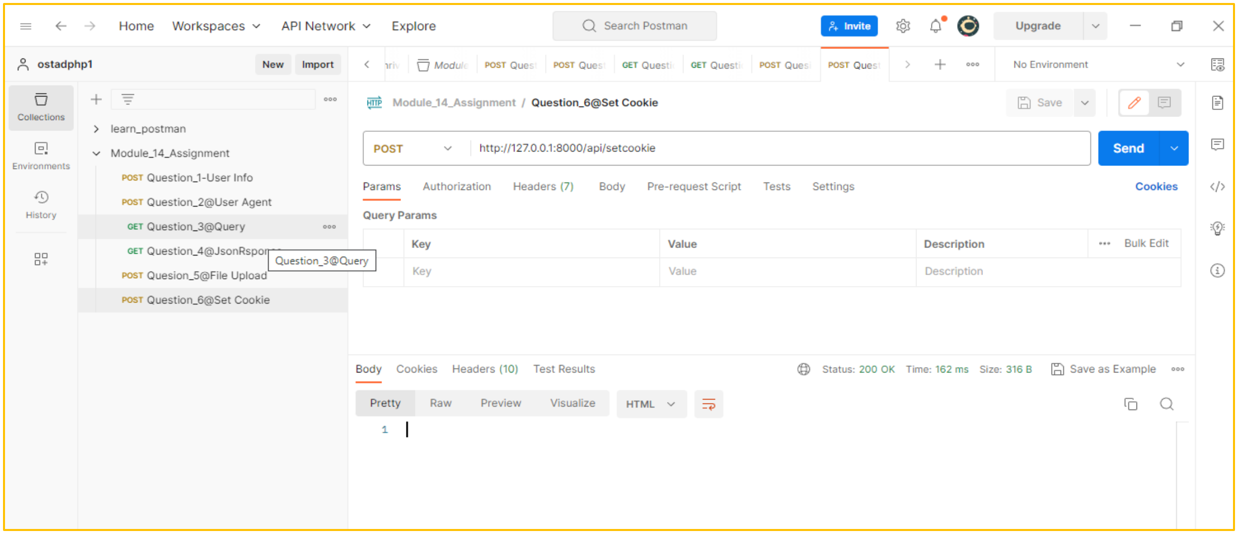


```
// Question-6 route
Route::post('/setcookie',[AssignmentController::class,'SetCookie']);
```



```
// Question-5 method
function FileUploads(Request $request):bool{
    $file=$request->file('avatar');
    $file->move(public_path('uploads'),$file->getClientOriginalName());
    return true;
}
```

Output(postman):



Module_14_Assignment / Question_6@Set Cookie

POST http://127.0.0.1:8000/api/setcookie

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Description
Key	Value	Description

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 162 ms Size: 316 B — Save as Example

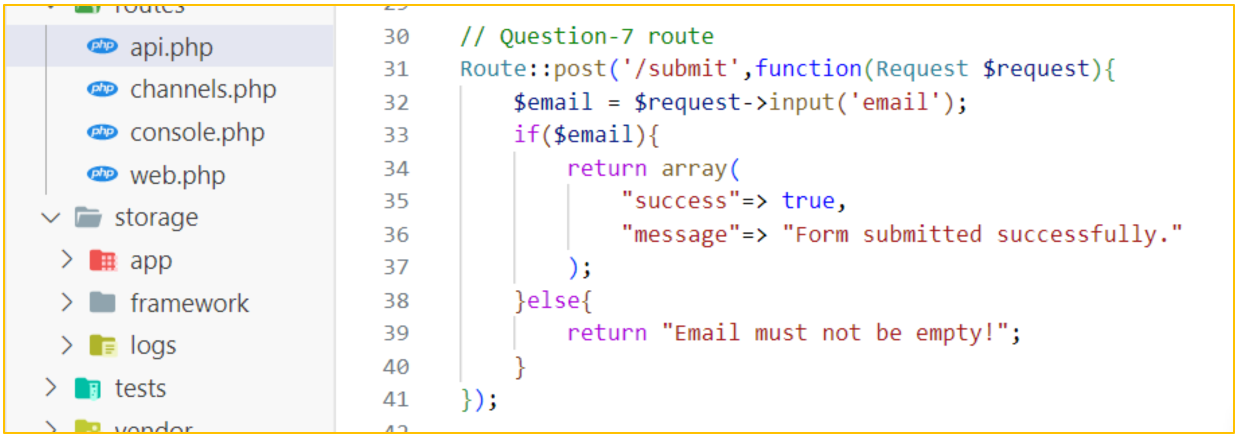
Pretty Raw Preview Visualize HTML

```
1
```

Question 7:

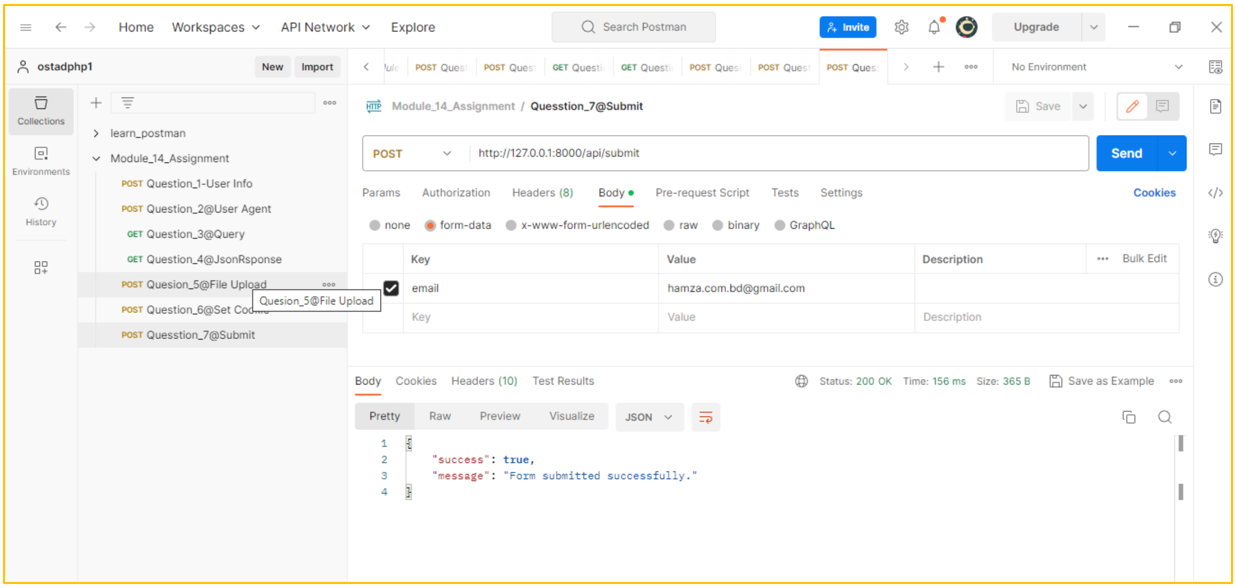
Question 7: Create a route in Laravel that handles a POST request to the '/submit' URL. Inside the route closure, retrieve the 'email' input parameter from the request and store it in a variable called $email. Return a JSON response with the following data:

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



```
// Question-7 route
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(postman):



Module_14_Assignment / Quesstion_7@Submit

POST http://127.0.0.1:8000/api/submit

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

○ none ● form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	Key	Value	Description
☑	email	hamza.com.bd@gmail.com	
	Key	Value	Description

Body Cookies Headers (10) Test Results — Status: 200 OK Time: 156 ms Size: 365 B — Save as Example

Pretty Raw Preview Visualize JSON

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