

**Assignment Module-14**  
**Name: Md. Shahidul Islam**  
**Course Name: PHP & Laravel Batch-1**  
**User ID: 01834348932**

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

```
// Question 1 api.php
Route::post('/userinfo',[AssignmentController::class,'UserInfo']);
```

Postman output:

The image shows the Postman application interface. On the left, a sidebar lists several API requests under the workspace 'Assignment(Module-14)'. The selected request is 'POST Question 1(user-info)'. The main panel shows the details of this request: a POST method to the URL 'http://127.0.0.1:8000/api/userinfo'. The 'Body' tab is active, showing a 'form-data' type with a single key-value pair: 'name' with the value 'Mazbaul Islam'. Below the request details, the response is displayed: a status of '200 OK', a response time of '1441 ms', and a body of 'Mazbaul Islam'.

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

Key	Value	Description

Body: 200 OK 1441 ms 328 B  
Mazbaul Islam

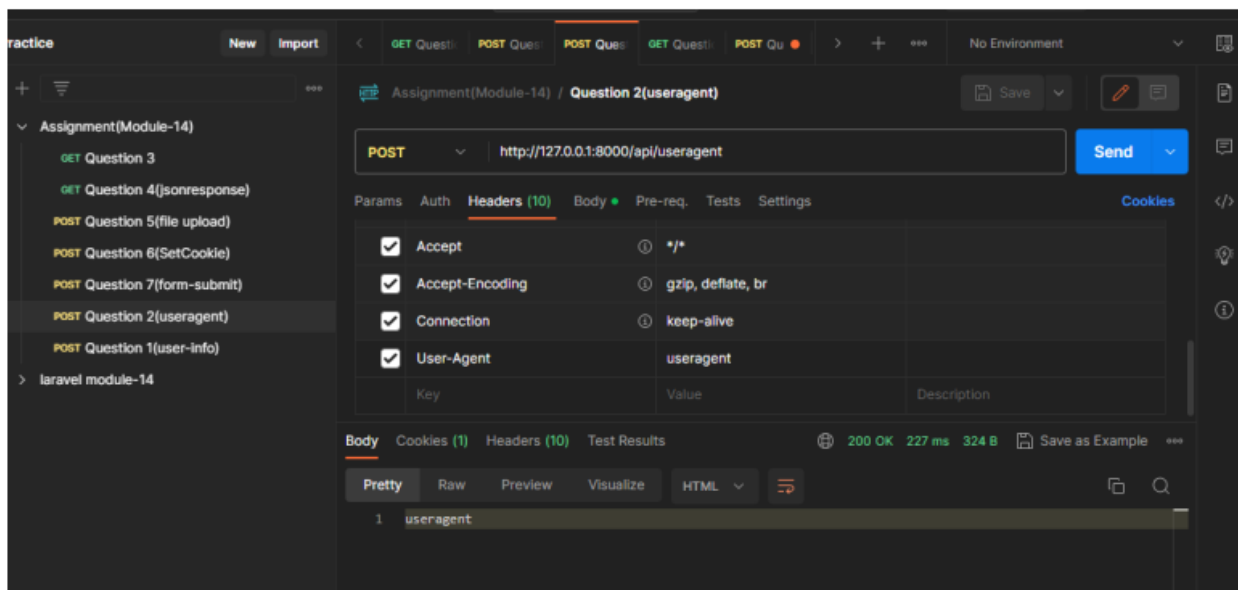
### Question 2:

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

```
// Question 2 api.php
Route::post('/useragent',[AssignmentController::class,'UserAgent']);
```

Postman output:



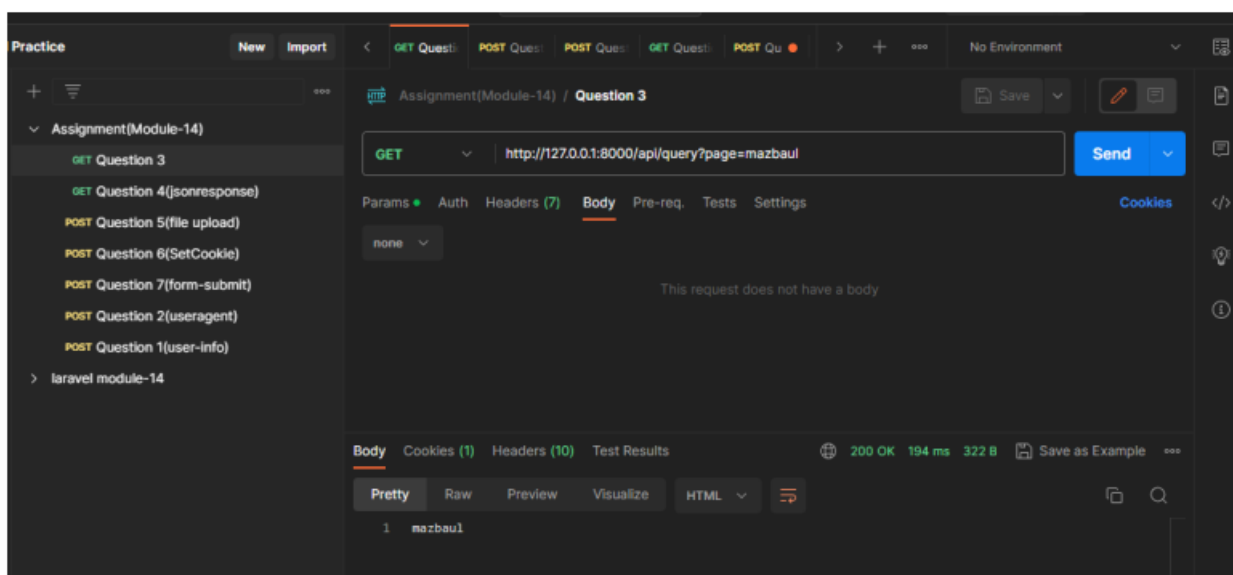
### Question 3:

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 AssignmentController.php
function ApiEndpoint(Request $request){
    $page = $request->query('page',null);
    if($page !== null){
        return $page;
    }else{
        return;
    }
}
```

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

Postman output:



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

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

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

```
// Question 4 api.php
Route::get('/jsonresponse',[AssignmentController::class,'JsonResponse']);
```

Postman output:

The screenshot shows the Postman interface for a GET request to `http://127.0.0.1:8000/api/jsonresponse`. The response is a 200 OK status with a response time of 209 ms and a body size of 364 B. The response body is displayed in JSON format:

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

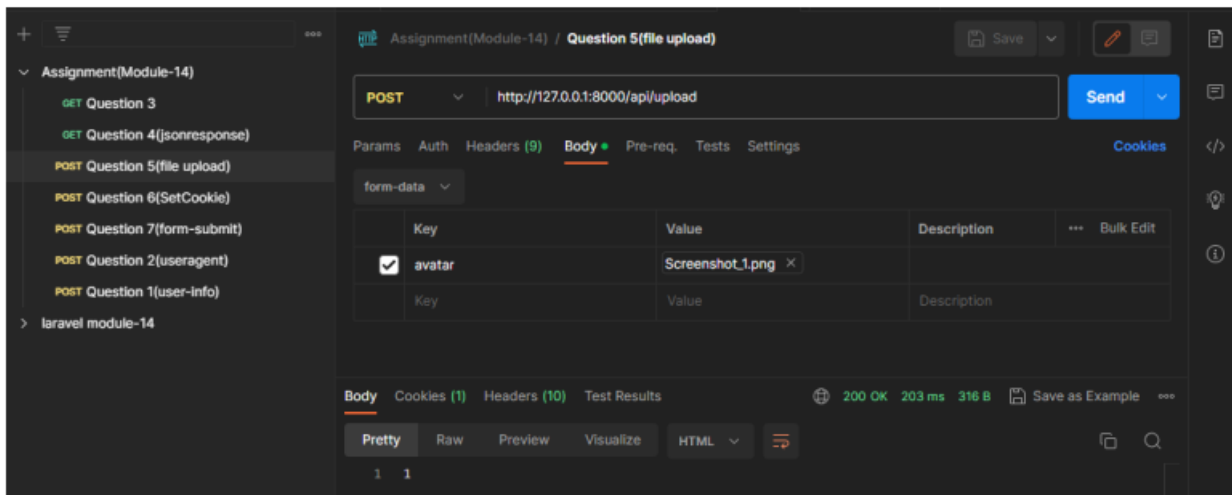
Question 5:

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

```
// Question 5 api.php
Route::post('/upload',[AssignmentController::class,'FileUploads']);
```

Postman output:



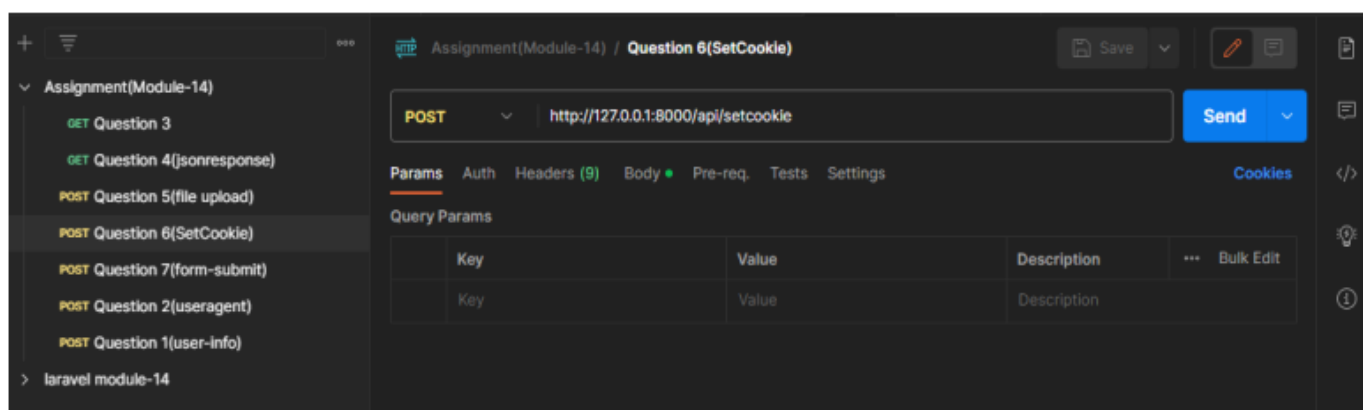
### Question 6:

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.

```
// Question 6 AssignmentController.php
function SetCookie(Request $request){
    $rememberToken = $request->cookie('remember_token',null);
    return $rememberToken;
}
```

```
// Question 6 api.php
Route::post('/setcookie',[AssignmentController::class,'SetCookie']);
```

### Postman Output:



### 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 api.php
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!";
    }
});
```

Postman output:

The image shows the Postman application interface. On the left, a sidebar lists several API requests under the folder 'Assignment(Module-14)'. The request 'POST Question 7(form-submit)' is selected and highlighted. The main panel displays the details of this request. At the top, it shows a POST request to the URL 'http://127.0.0.1:8000/api/submit'. Below this, the 'Body' tab is active, showing a 'form-data' type with a single key-value pair: 'email' with the value 'mazbaul@gmail.com'. At the bottom, the 'Body' tab shows the response in 'Pretty' format, which is a JSON object: {'success': true, 'message': 'Form submitted successfully.'}. The status bar at the bottom indicates a 200 OK response with a response time of 195 ms and a body size of 364 B.

Assignment(Module-14) / Question 7(form-submit)

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

form-data

Key	Value	Description
email	mazbaul@gmail.com	

Body Cookies (1) Headers (10) Test Results

200 OK 195 ms 364 B Save as Example

Pretty Raw Preview Visualize JSON

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