

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

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

Postman output:

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection named 'Assignment14' is expanded, showing a request named 'GET Question1'. The request is a POST to 'http://127.0.0.1:8000/api/userinfo?name=Nahid'. The 'Query Params' section shows a single parameter 'name' with the value 'Nahid'. The 'Body' tab is selected, showing the response 'Nahid'. The status is '200 OK' with a response time of '135 ms' and a body size of '320 B'.

Key	Value	Description
name	Nahid	

Key	Value	Description

Body: Nahid

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

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

The screenshot displays the Postman application interface. On the left, the 'My Workspace' sidebar shows a collection named 'Assignment14' containing two requests: 'GET Question1' and 'GET Question2'. The main panel shows a 'POST' request to 'http://127.0.0.1:8000/api/useragent'. The 'Headers' tab is active, showing a table of headers:

Key	Value	Description
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>	
<input checked="" type="checkbox"/> Content-Length	0	Go to body
<input checked="" type="checkbox"/> Host	<calculated when request is sent>	
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.32.2	
<input checked="" type="checkbox"/> Accept	*/*	
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br	

Below the headers, the 'Body' tab is selected, showing the response in 'Pretty' format: '1 PostmanRuntime/7.32.2'. The status bar at the bottom indicates a '200 OK' response with a time of '4.25 s' and a size of '336 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
function ApiPoint(Request $request){
    $page = $request->query('page',null);
    if($page !== null){
        return $page;
    }else{
        return;
    }
}
```

```
// Question 3
Route::get('/query',[newcontro::class,'ApiPoint']);
```

Postman Output:

The screenshot shows the Postman interface with a GET request to `http://127.0.0.1:8000/api/query?page=Nahid`. The request is saved under the collection 'Assignment14' with the name 'Question3'. The 'Query Params' section shows a parameter 'page' with the value 'Nahid'. The 'Body' tab is selected, showing the response 'Nahid'.

Key	Value	Description
<input checked="" type="checkbox"/> page	Nahid	
Key	Value	Description

Response: 200 OK, 168 ms, 320 B. The response body is 'Nahid'.

Question 4:

Create a JSON response in Laravel with the following data:

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

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

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

Postman Output:

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection 'Assignment14' is expanded, showing 'Question4'. The request is a GET to 'http://127.0.0.1:8000/api/jsonresponse'. The response is a JSON object:

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

. The status bar indicates '200 OK', '321 ms', and '364 B'.

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

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

Postman Output:

The screenshot shows the Postman interface with a workspace named 'My Workspace'. A collection named 'Assignment14' is expanded, showing several GET requests. The selected request is 'Question5' with the URL 'http://127.0.0.1:8000/api/upload'. The request is a GET method. The 'Body' tab is selected, showing a table with one row: 'image' with value 'Screenshot_8.png'. The response is displayed at the bottom, showing a status of '200 OK', a latency of '116 ms', and a body of '1'.

Key	Value	Description
<input checked="" type="checkbox"/> image	Screenshot_8.png	
Key	Value	Description

Body	Cookies	Headers (10)	Test Results
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
function Cookies(Request $request){
    $rememberToken = $request->cookie('remember_token',null);
    return $rememberToken;
}
```

```
// Question 6
Route::post('/setcookie',[newcontro::class,'Cookies']);
```

The screenshot displays the Postman API client interface. The left sidebar shows a workspace named 'My Workspace' with a collection 'Assignment14' containing several GET requests. The main panel shows a POST request named 'Question6' to the URL 'http://127.0.0.1:8000/api/setcookie'. The 'Params' tab is active, showing a table for query parameters. The 'Body' tab is also visible, showing a status of 200 OK.

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

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

Query Params

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (10) Test Results

Status: 200 OK Time: 169 ms Size: 315 B Save as Example

Pretty Raw Preview Visualize HTML

1

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!";
    }
});
```

HomeWorkspacesAPI NetworkExplore

Search Postman

Invite

Upgrade

My Workspace

NewImport

GET New RPOST QuGET QueGET QuePOST QuPOST QuPOST Qu

No Environment

Collections

Assignment14

GET Question1GET Question2GET Question3GET Question4GET Question5GET Question6GET Question7

Question2question2

Assignment14 / Question7

Save

Send

POST

http://127.0.0.1:8000/api/submit?email=mdnahidhossencse@gmail.com

ParamsAuthorizationHeaders (8)BodyPre-request ScriptTestsSettings

Cookies

Query Params

	Key	Value	Description	Bulk Edit
<input checked="" type="checkbox"/>	email	mdnahidhossencse@gmail.com		
	Key	Value	Description	

BodyCookiesHeaders (10)Test Results

Status: 200 OKTime: 159 msSize: 364 BSave as Example

PrettyRawPreviewVisualizeJSON

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

OnlineFind and replaceConsole

RunnerCapture requestsCookiesTrash