Course name: COMS 319

Date: 4/10/24

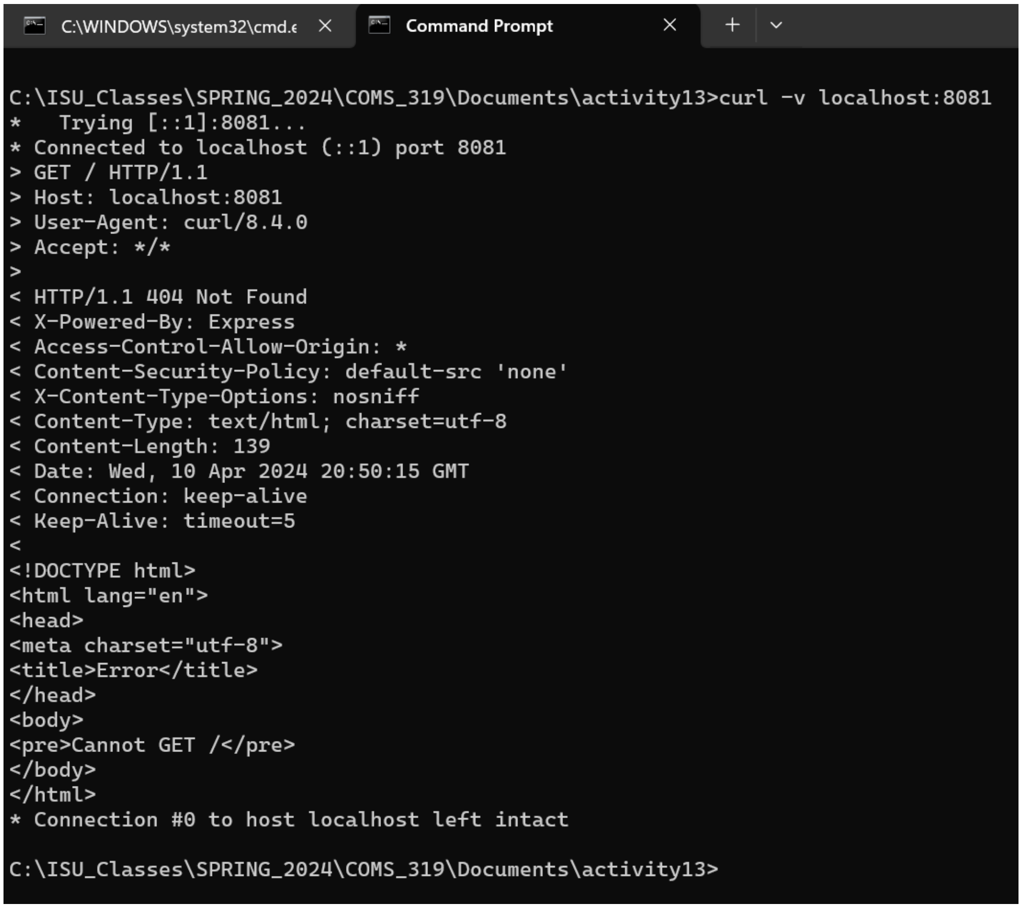
Student name: Isabelle Singh

Student email: bellas23@iastate.edu

REST application – From my understanding, a REST application is something that two computers use to transfer information securely on the internet. REST stands for REpresentational State Transfer.

API – API stands for Application Programming Interface. It is used to create rules that other systems will follow in order to communicate.

"curl -v localhost:8081/":

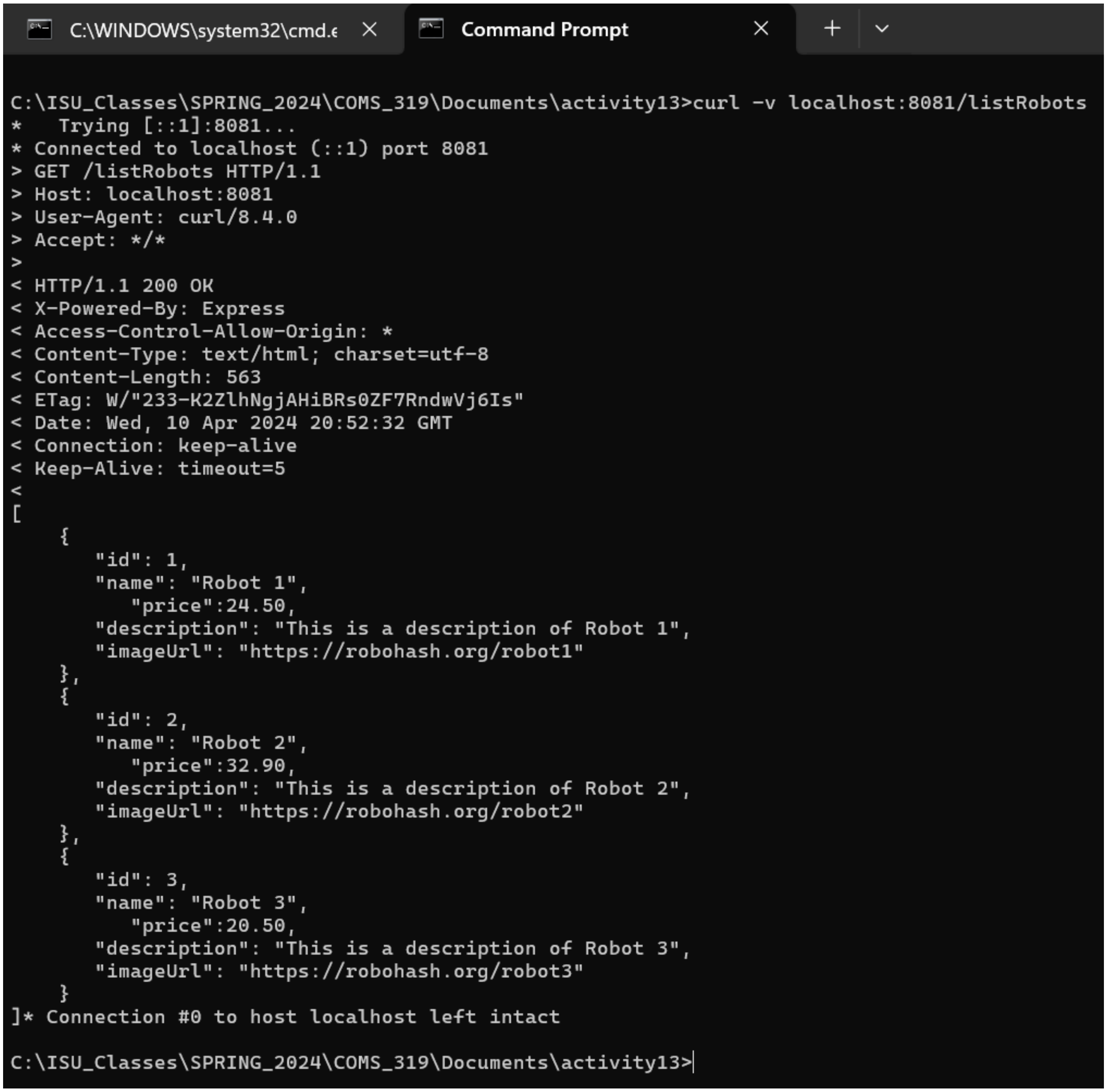


```
C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>curl -v localhost:8081
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET / HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 404 Not Found
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Security-Policy: default-src 'none'
< X-Content-Type-Options: nosniff
< Content-Type: text/html; charset=utf-8
< Content-Length: 139
< Date: Wed, 10 Apr 2024 20:50:15 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>Cannot GET /</pre>
</body>
</html>
* Connection #0 to host localhost left intact

C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>
```

The connection could not be made because the server responded with a status code of 404 Not Found, meaning that the request was made for a resource that does not exist on the server. Essentially, the server received the GET request for the root URL ("/"), but it couldn't find a resource to fulfill that request, resulting in a 404 Not Found error.

“curl -v localhost:8081/listRobots” :



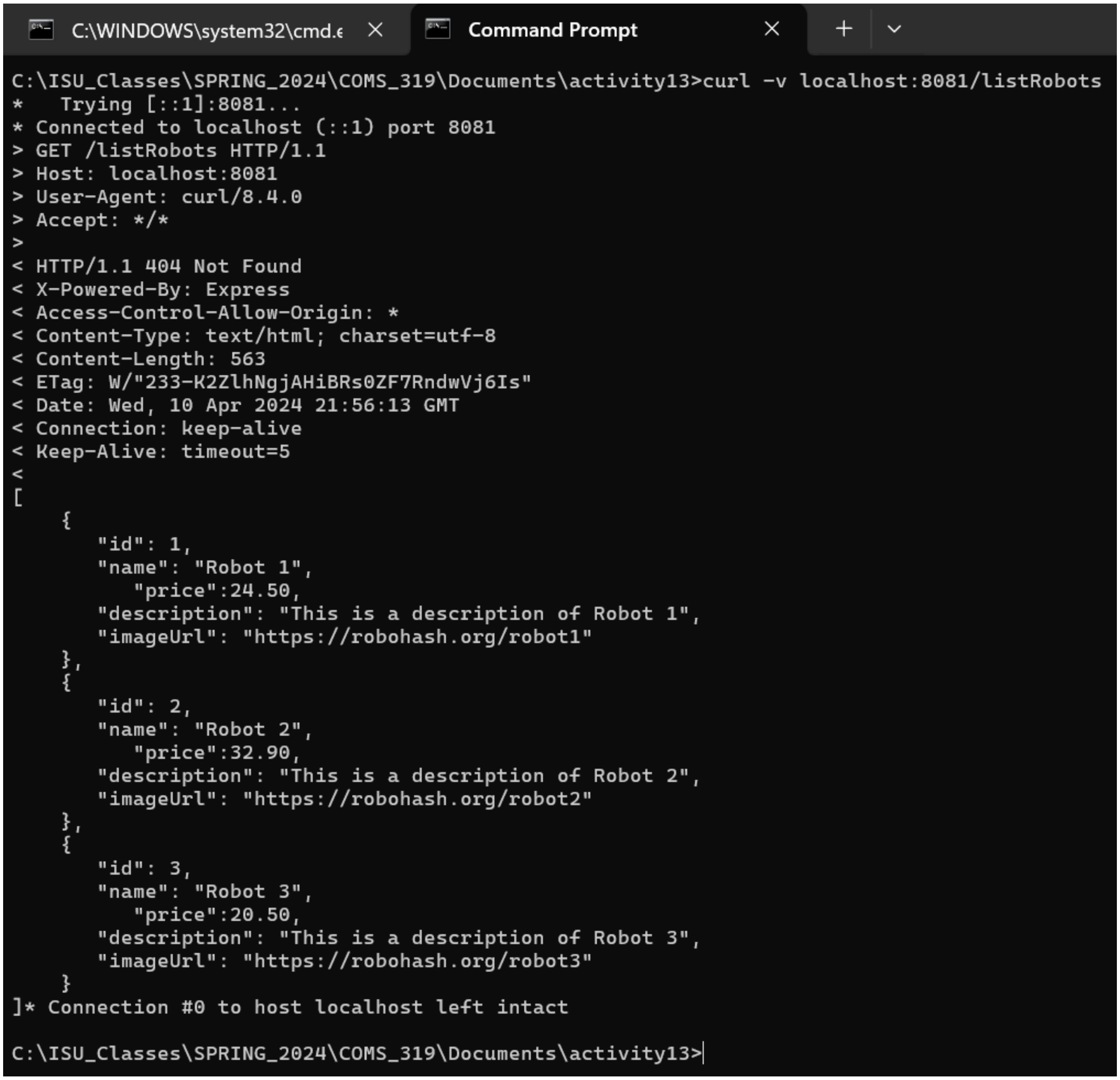
```
C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>curl -v localhost:8081/listRobots
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET /listRobots HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 563
< ETag: W/"233-K2ZlhNgjAHiBRs0ZF7RndwVj6Is"
< Date: Wed, 10 Apr 2024 20:52:32 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[
    {
       "id": 1,
       "name": "Robot 1",
          "price":24.50,
       "description": "This is a description of Robot 1",
       "imageUrl": "https://robohash.org/robot1"
    },
    {
       "id": 2,
       "name": "Robot 2",
          "price":32.90,
       "description": "This is a description of Robot 2",
       "imageUrl": "https://robohash.org/robot2"
    },
    {
       "id": 3,
       "name": "Robot 3",
          "price":20.50,
       "description": "This is a description of Robot 3",
       "imageUrl": "https://robohash.org/robot3"
    }
]* Connection #0 to host localhost left intact

C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>
```

Request: The curl command sends a GET request to the server running on localhost at port 8081. The request method is specified as GET, which indicates that the client is asking the server to retrieve data from a specified resource. The HTTP version used is HTTP/1.1, which is a common version of the HTTP protocol. The request includes headers such as Host, User-Agent, and Accept, which provide information about the client and its preferences to the server.

Response: The server responds with an HTTP/1.1 200 OK status code. This status code indicates that the request was successfully accepted by the server and the server is now sending back the requested data. The Content-Type header specifies that the content of the response is text/html; charset=utf-8, but the actual content returned is a JSON array of robot objects.

“curl -v localhost:8081/listRobots” after changing the status code to 404 in the App.js:

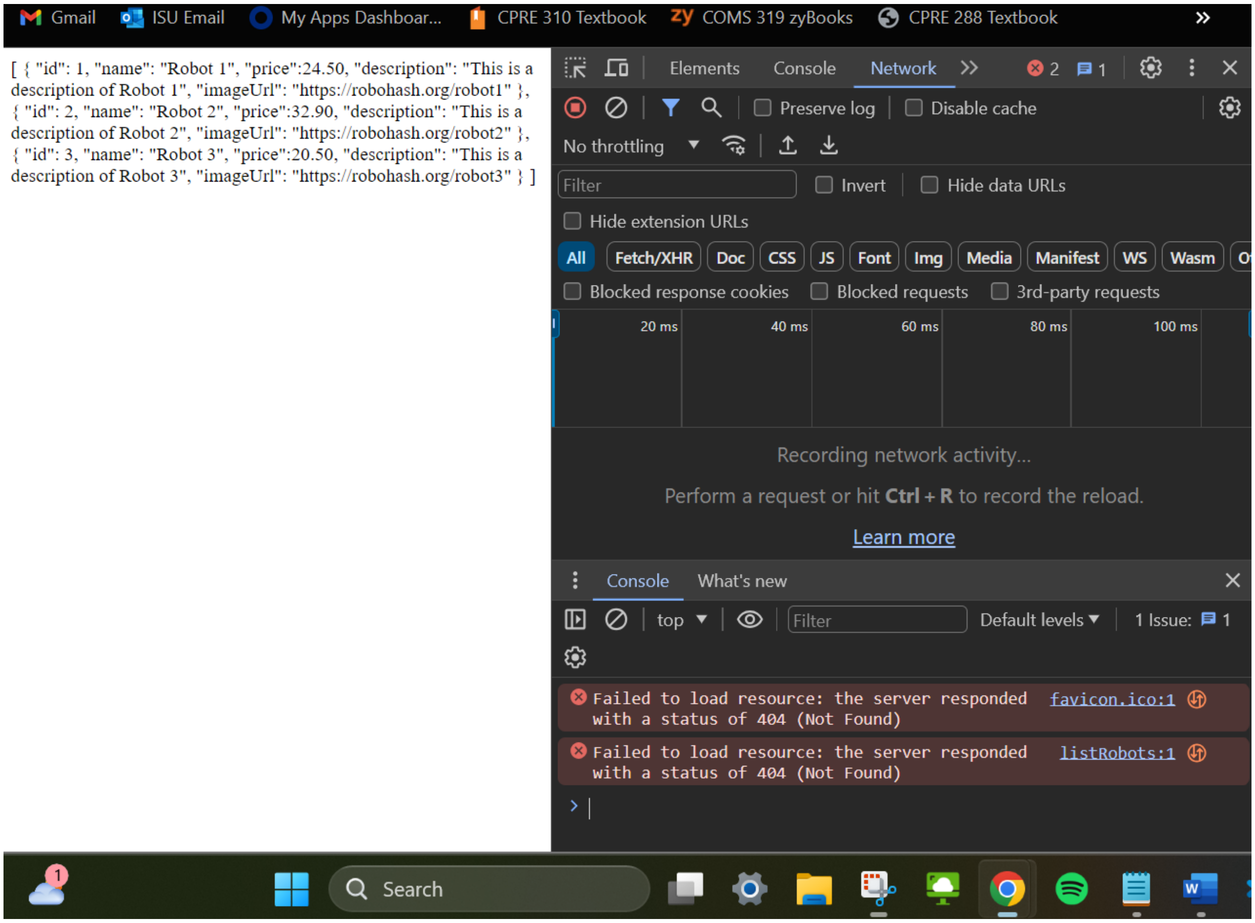


```
C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>curl -v localhost:8081/listRobots
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET /listRobots HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 404 Not Found
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 563
< ETag: W/"233-K2ZlhNgjAHiBRs0ZF7RndwVj6Is"
< Date: Wed, 10 Apr 2024 21:56:13 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
[
    {
       "id": 1,
       "name": "Robot 1",
          "price":24.50,
       "description": "This is a description of Robot 1",
       "imageUrl": "https://robohash.org/robot1"
    },
    {
       "id": 2,
       "name": "Robot 2",
          "price":32.90,
       "description": "This is a description of Robot 2",
       "imageUrl": "https://robohash.org/robot2"
    },
    {
       "id": 3,
       "name": "Robot 3",
          "price":20.50,
       "description": "This is a description of Robot 3",
       "imageUrl": "https://robohash.org/robot3"
    }
]* Connection #0 to host localhost left intact

C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>
```

The status code 404 means the requested resource could not be found on the server. It is part of the 4XX class of status codes, which are client error responses. The response body still contains data about robots, but the status code indicates that the /listRobots endpoint or resource is not found.

http://localhost:8081/listRobots after changing the status code in the App.js:

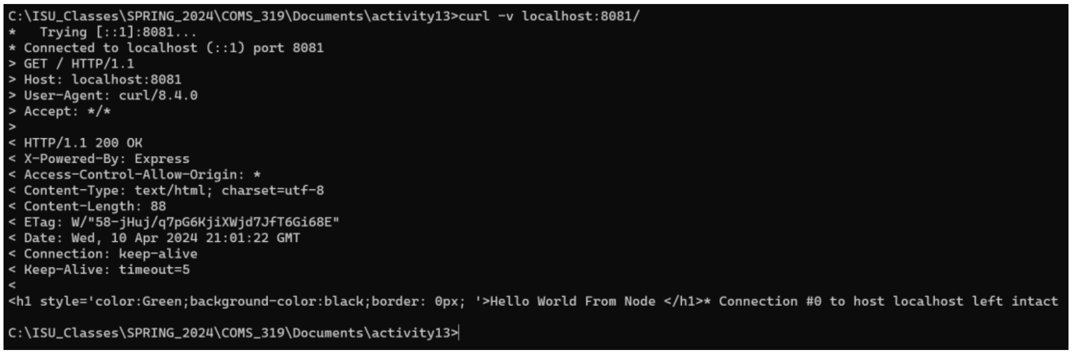


```
[ { "id": 1, "name": "Robot 1", "price":24.50, "description": "This is a description of Robot 1", "imageUrl": "https://robohash.org/robot1" }, { "id": 2, "name": "Robot 2", "price":32.90, "description": "This is a description of Robot 2", "imageUrl": "https://robohash.org/robot2" }, { "id": 3, "name": "Robot 3", "price":20.50, "description": "This is a description of Robot 3", "imageUrl": "https://robohash.org/robot3" } ]
```

```
Failed to load resource: the server responded    favicon.ico:1
with a status of 404 (Not Found)
Failed to load resource: the server responded    listRobots:1
with a status of 404 (Not Found)
```

The corresponding request is highlighted in red to indicate a client-side error. It says, “Failed to load resource” and displays "404 Not Found."

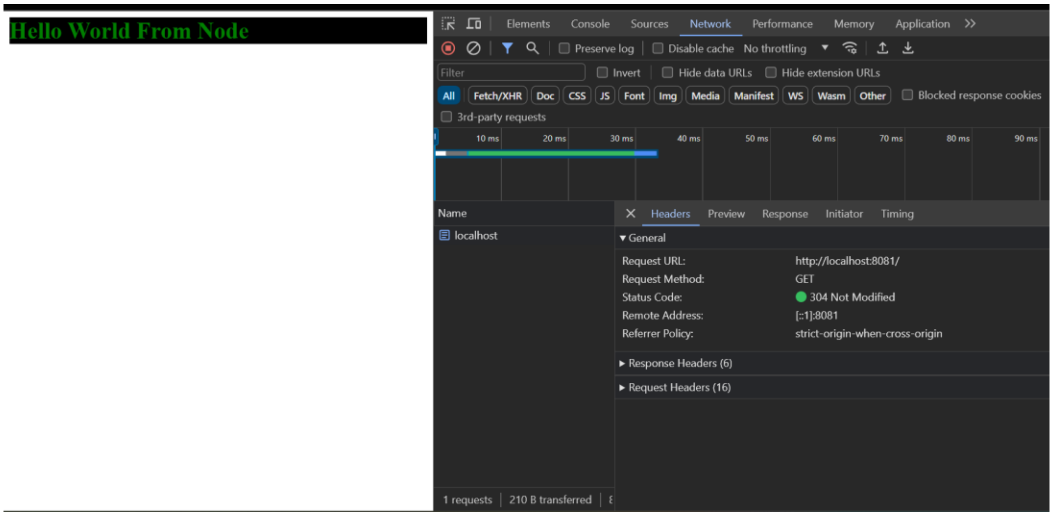
“curl -v localhost:8081/” after changing App.js with HTML code:



```
C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>curl -v localhost:8081/
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET / HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: text/html; charset=utf-8
< Content-Length: 88
< ETag: W/"58-jHuj/q7pG6KjiXWjd7JfT6Gi68E"
< Date: Wed, 10 Apr 2024 21:01:22 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
<h1 style='color:Green;background-color:black;border: 0px; '>Hello World From Node </h1>* Connection #0 to host localhost left intact

C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>
```

“http://localhost:8081/” after changing App.js with HTML code:



Hello World From Node

▾ General

Request URL: http://localhost:8081/
Request Method: GET
Status Code: ● 304 Not Modified
Remote Address: [::1]:8081
Referrer Policy: strict-origin-when-cross-origin

▸ Response Headers (6)

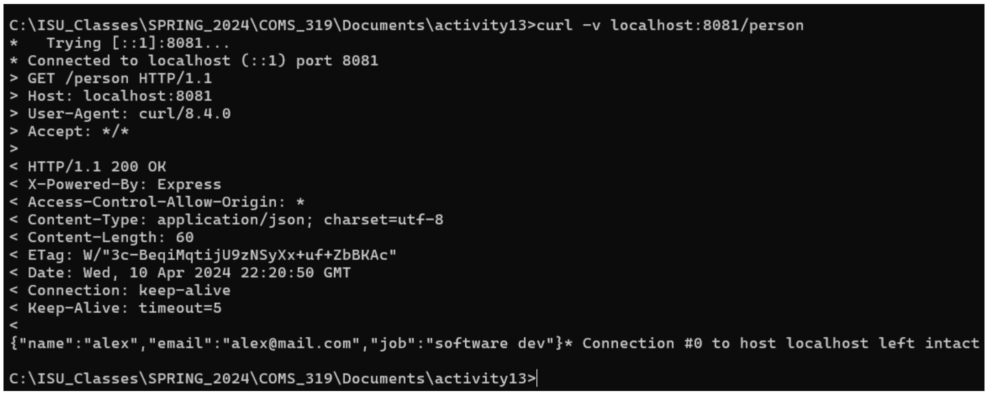
▸ Request Headers (16)

1 requests | 210 B transferred

The response in the command line or terminal indicates that the request to localhost:8081/ was successful, with a status code of 200 OK. The server responded with the specified HTML code, including a green-colored "Hello World From Node" header. The response headers include information such as the server being powered by Express, allowing cross-origin requests, and specifying the content type as HTML.

In the browser, you can see the rendered HTML, displaying the "Hello World From Node" header in green color as specified in the HTML code. The status code of 200 OK indicates that the request was successful, and the browser interprets the HTML response to render it for the user.

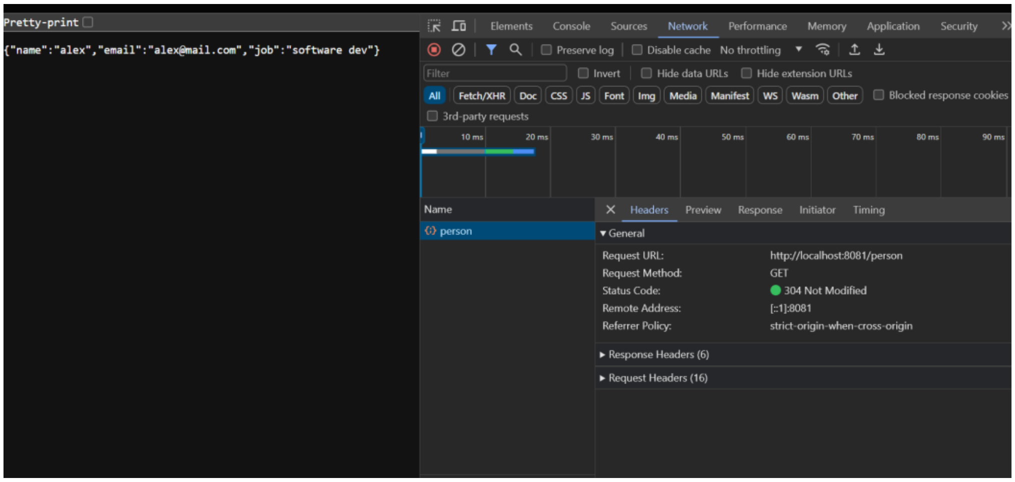
“curl -v localhost:8081/person” after changing App.js to respond with a JavaScript Object code:



```
C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>curl -v localhost:8081/person
*   Trying [::1]:8081...
* Connected to localhost (::1) port 8081
> GET /person HTTP/1.1
> Host: localhost:8081
> User-Agent: curl/8.4.0
> Accept: */*
>
< HTTP/1.1 200 OK
< X-Powered-By: Express
< Access-Control-Allow-Origin: *
< Content-Type: application/json; charset=utf-8
< Content-Length: 60
< ETag: W/"3c-BeqiMqtijU9zNSyXx+uf+ZbBKAc"
< Date: Wed, 10 Apr 2024 22:20:50 GMT
< Connection: keep-alive
< Keep-Alive: timeout=5
<
{"name":"alex","email":"alex@mail.com","job":"software dev"}* Connection #0 to host localhost left intact

C:\ISU_Classes\SPRING_2024\COMS_319\Documents\activity13>
```

“localhost:8081/person” after changing App.js to respond with a JavaScript Object code:



Pretty-print □

```
{"name":"alex","email":"alex@mail.com","job":"software dev"}
```

□ Preserve log □ Disable cache No throttling

□ Invert □ Hide data URLs □ Hide extension URLs

□ Blocked response cookies

□ 3rd-party requests

Request URL: http://localhost:8081/person
Request Method: GET
Status Code: ● 304 Not Modified
Remote Address: [::1]:8081
Referrer Policy: strict-origin-when-cross-origin

The server responds with a JSON object containing information about a person, including their name, email, and job. The response headers indicate that the content type is application/json, and the response status is 200 OK, indicating success. The response includes other headers such as X-Powered-By: Express and Access-Control-Allow-Origin: *. The content of the response body is the JSON object representing the person.

In the web browser's Network section, when accessing localhost:8081/person, the request is sent to the server, and the response is received, similarly containing the JSON object with the person's information. The content type is also application/json and shows the request and response headers along with the response body.