Computer Network

Honorary Project : AJAX Web

Report

COMPUTER NETWORK PROJECT

Subject	Computer Network
Professor	Jin Seek Choi
Submission Date	December 8th, 2021
University	Hanyang University
School	College of Engineering
Department	Department of Computer Science & Engineering
Student ID	**Name**
2019009261	최가온(CHOI GA ON)

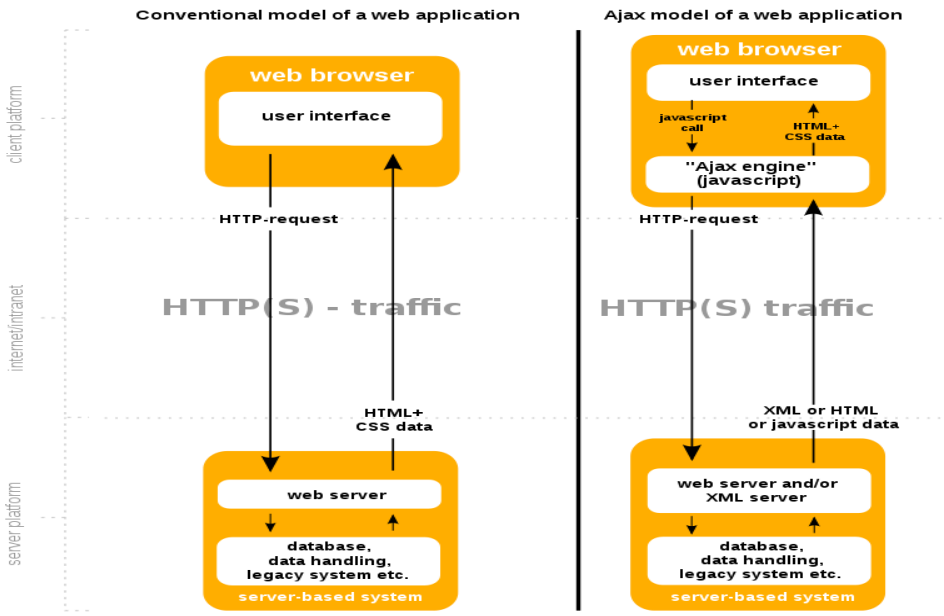
Project Preview

AJAX stands for “Asynchronous JavaScript and XML”. AJAX is one of the development techniques for creating fast-acting dynamic web pages. It can only update a part of the web page without reloading the entire web page. In other words, AJAX allows us to communicate with the server in the background area and display the results only on a part of the web page.

The techniques used in this project to implement AJAX-based web page are as follows:

- HTML for web page representation
- XMLHttpRequest object for asynchronous communication with a web page server
- JavaScript for controlling the user’s workflow by combining all the technologies mentioned above.

The figure shown below describes the differences between traditional HTTP communication methods and AJAX-based communication methods used in this project.



The detailed process of communication using AJAX is as follows:

① A request event occurs by the user.

② When a request event occurs, JavaScript is called by the event handler.

③ JavaScript sends a request to the server using the XMLHttpRequest object. At this time, the web browser can process other tasks without having to wait for the server's response after sending the request.

④ The server processes AJAX requests with the XMLHttpRequest object that is delivered.

⑤ The server delivers the processed results to the web browser as HTML, XML, or JSON data.

⑥ The responses delivered at this time do not send all new pages, but only deliver the necessary data.

⑦ Call JavaScript to update only a portion of the web page with the data received from the server.

⑧ As a result, only a part of the web server is reloaded and displayed.

Code Explanation

1. main.html

Step 1.

First, the title and usage of the web page are displayed in the “div class=Main” part.

```
<html>
    <head>

    </head>
    <body>
        <div class="Main">
            <h1>AJAX Project: Personal Web Page</h1>
            <h2>Hello, I'm Gaon Choi.</h2>
            <h3>Press the buttons below to view details.</h3>
        </div>
```

Step 2.

Create button objects to be displayed on the web page, and hand functions as on-click events on each button. And create the “div id=content” part. Each time a button is pressed, the contents of the HTML file to be loaded are brought to this location.

```
        <form>
            <input onclick="get_contactinfo();" type="button" value="Contact Info" />
            <input onclick="get_eduinfo();" type="button" value="Education" />
            <input onclick="get_skillinfo();" type="button" value="Skills" />
            <input onclick="get_activityinfo();" type="button" value="Activities" />
            <input onclick="get_awardinfo();" type="button" value="Awards" />
            <input onclick="get_presentationinfo();" type="button" value="Presentations" />
            <input onclick="get_imageinfo();" type="button" value="Gallery" />
            <input onclick="hidecontent();" type="button" value="HIDE CONTENTS"
             style="background-color:skyblue" />
        </form>

        <div id="content">blank</div>
```

Step 3.

From here on, it is about JavaScript section. Functions to be used in the web page are implemented one by one. The get_contactinfo function brings up the contact.html file. And put the contents of the file into .innerHTML so that contact information is displayed on the actual web page.

```
<script type="text/javascript">
            async function get_contactinfo() {
                var request = new XMLHttpRequest();
                await request.open("GET", "./src/contact.html", true);
                await request.send();

                request.onload = function () {
                    console.log('!!', request.responseText);
                    console.log('!!', document.getElementById("content").innerHTML);
                    document.getElementById("content").innerHTML = request.responseText;
                }
            }
```

Step 4.

The get_eduinfo function brings up the education.html file. And put the contents of the file into .innerHTML so that educational institution information is displayed on the actual web page.

```
            async function get_eduinfo() {
                var request = new XMLHttpRequest();
                await request.open("GET", "./src/education.html", true);
                await request.send();

                request.onload = function() {
                    console.log('!!', request.responseText);
                    console.log('!!', document.getElementById("content").innerHTML);
                    document.getElementById("content").innerHTML = request.responseText;
                }
            }
```

Step 5.

The get_skillinfo function brings up the skills.html file. And put the contents of the file into .innerHTML so that a list of technologies that the author can use, is displayed on the actual web page.

```
async function get_skillinfo() {
    var request = new XMLHttpRequest();
    await request.open("GET", "./src/skills.html", true);
    await request.send();

    request.onload = function() {
        console.log('!!', request.responseText);
        console.log('!!', document.getElementById("content").innerHTML);
        document.getElementById("content").innerHTML = request.responseText;
    }
}
```

Step 6.

The get_activityinfo function brings up the activities.html file. And put the contents of the file into .innerHTML so that author's career activity information is displayed on the actual web page.

```
async function get_activityinfo() {
    var request = new XMLHttpRequest();
    await request.open("GET", "./src/activities.html", true);
    await request.send();

    request.onload = function() {
        console.log('!!', request.responseText);
        console.log('!!', document.getElementById("content").innerHTML);
        document.getElementById("content").innerHTML = request.responseText;
    }
}
```

Step 7.

The get_awardinfo function brings up the award.html file. And put the contents of the file into .innerHTML so that author's awards information is displayed on the actual web page.

```
async function get_awardinfo() {
    var request = new XMLHttpRequest();
    await request.open("GET", "./src/award.html", true);
    await request.send();

    request.onload = function() {
        console.log('!!', request.responseText);
        console.log('!!', document.getElementById("content").innerHTML);
        document.getElementById("content").innerHTML = request.responseText;
    }
}
```

Step 8.

The get_presentationinfo function brings up the presentation.html file. And put the contents of the file into .innerHTML so that author's official presentation information is displayed on the actual web page.

```
async function get_presentationinfo() {
    var request = new XMLHttpRequest();
    await request.open("GET", "./src/presentation.html", true);
    await request.send();

    request.onload = function() {
        console.log('!!', request.responseText);
        console.log('!!', document.getElementById("content").innerHTML);
        document.getElementById("content").innerHTML = request.responseText;
    }
}
```

Step 9.

The get_imageinfo function brings up the image.png file. And put the contents of the file into .innerHTML so that graphical information is displayed on the actual web page.

```
            async function get_imageinfo() {
                var image = new Image();
                var div = document.getElementById("content");
                div.innerHTML="";
                image.onload = function(){
                    div.appendChild(image);
                };
                image.src = "./src/image.png";
                image.height="400";
            }
```

Step 10.

The hidecontent function serves to erase the content printed on the web page by marking the .innerHTML segment 'blank'.

```
            function hidecontent() {
                var element = document.getElementById("content");
                element.innerHTML = "";
            }
        </script>
    </body>
</html>
```

2. contact.html

The entire code of contact.html is as follows:

```
<ul>
    <h3>E-mail</h3>
    <h4 style="font-weight: lighter;">choigaon1028@hanyang.ac.kr</h4>

    <h3>Github</h3>
    <h4 style="font-weight: lighter;">https://github.com/Gaon-Choi</h4>

    <h3>LikedIn</h3>
    <h4 style="font-weight: lighter;">https://www.linkedin.com/in/gaon-choi</h4>
</ul>
```

3. education.html

The entire code of education.html is as follows:

```
<ul>
    <h3>Hanyang University(Seoul)</h3>
    <h4 style="font-weight: lighter; color:blueviolet;">Department of Computer Science and Engineering</h4>
    <h5 style="font-weight: lighter;">2019.03.01. ~ PRESENT </h5>


    <h4 style="font-weight: lighter; color:blue">Department of Electronic Engineering</h4>
    <h5 style="font-weight: lighter;">2020.09.01. ~ PRESENT </h5>
    <h5 style="font-weight: lighter;">Double major </h5>
    <h5 style="font-weight: lighter;">Interested in : Signal Processing, Semiconductor </h5>


    <h3>Gwangju High School</h3>
    <h4 style="font-weight: lighter;">2016.03.01. ~ 2019.02.28.</h4>


    <h3>Duam Middle School</h3>
    <h4 style="font-weight: lighter;">2013.03.01. ~ 2016.02.28.</h4>


    <h3>Yul-gok Elementary School</h3>
    <h4 style="font-weight: lighter;">2007.03.01. ~ 2013.02.28.</h4>
</ul>
```

4. skills.html

The entire code of skills.html is as follows:

```
<ul>
    <h3>Programming Languages</h3>
    <h4 style="font-weight: bolder;">Python</h4>
    <h4 style="font-weight: lighter;">Machine Learning: sklearn, pytorch</h4>
    <h4 style="font-weight: lighter;">Data Science: matplotlib</h4>
    <h4 style="font-weight: lighter;">Computer Vision: Image, OpenCV</h4>
    <h4 style="font-weight: lighter;">Database: psycopg2</h4>

    <h4 style="font-weight: bolder;">C/C++</h4>
    <h4 style="font-weight: lighter;">for Embedded System development</h4>

    <h4 style="font-weight: bolder;">Java</h4>
    <h4 style="font-weight: bolder;">SQL</h4>
    <h4 style="font-weight: lighter;">for Database System</h4>
    <h4 style="font-weight: bolder;">MATLAB</h4>
    <h4 style="font-weight: lighter;">for Signal Processing</h4>

    <h3>Communation Tools</h3>
    <h4 style="font-weight: lighter;">Github, Slack, Discord</h4>

    <h3>In studying process ...</h3>
    <h4 style="font-weight: lighter;">LaTex, HTML, Streamlit</h4>
</ul>
```

5. activities.html

The entire code of activities.html is as follows:

```
<ul>
    <h3>Visual Computing Lab.</h3>
    <h4 style="font-weight: lighter;">Supervisor: Ph.D, Je Hyeong Hong</h4>
    <h4 style="font-weight: lighter;">Project: Body type data visualization demo service
planning research service</h4>

    <h3>SuresoftTech</h3>
    <h4 style="font-weight: lighter;">Department of CS-Static</h4>
    <h4 style="font-weight: lighter;">Winter vacation internship program (2020.12.28. ~
2021.02.28.)</h4>
    <h4 style="font-weight: lighter;">- Code rule analysis: MISRA C 2012, MISRA CPP
2008</h4>
```

```
    <h4 style="font-weight: lighter;">- Manual improvement: detection of  grammatical errors, suggestion of improved good/bad cases, translation of references</h4>

    <h3>Samsung Dream Class Winter Camp</h3>
    <h4 style="font-weight: lighter;">Conducted mathematics class and SW mentoring for middle school students during 2020 January, in Hanyang University at Ansan(ERICA Campus).</h4>
</ul>
```

6. award.html

The entire code of award.html is as follows:

```
<ul>
    <h3>Undergraduate Scholarship</h3>
    <h4 style="font-weight: lighter;">Hanyang Univ. CSE&EE Alumni Association</h4>

    <h3>Academic Honor Award</h3>
    <h4 style="font-weight: lighter;">2021 1st semester</h4>

    <h3>Taesul Scholarship foundation</h3>
    <h4 style="font-weight: lighter;">2021 1st semester</h4>

    <h3>Mirae Asset Science and Engineering Talent Scholarship</h3>
    <h4 style="font-weight: lighter;">2020, Mirae Asset</h4>

    <h3>Academic Honor Award</h3>
    <h4 style="font-weight: lighter;">2020 1st semester</h4>

    <h3>Academic Excellence Award</h3>
    <h4 style="font-weight: lighter;">2019 1st semester</h4>
</ul>
```

7. presentation.html

The entire code of presentation.html is as follows:

```
<ul>
    <h3>HAI(Hanyang Artificial Intelligence) lecturer</h3>
    <h4 style="font-weight: lighter;"> The basics of machine learning</h4>
    <h4 style="font-weight: lighter;">Book: Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow</h4>

    <h4 style="font-weight: lighter;">(sample) numpy, matplotlib, pandas</h4>
```

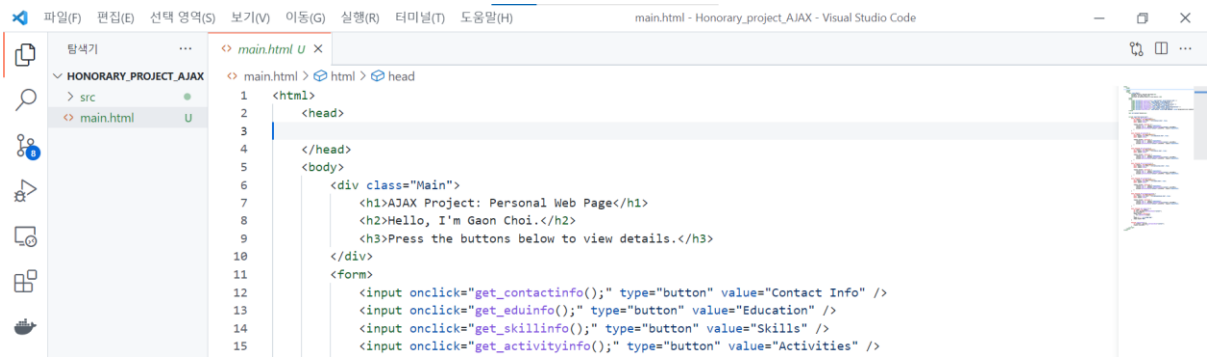
```
    <h5 style="font-weight: lighter;">https://www.youtube.com/watch?v=g97wzgi-ApM</h5>

    <h4 style="font-weight: lighter;">Github Repository</h4>
    <h5 style="font-weight: lighter;">https://github.com/Gaon-Choi/2021_HAI_Hands-On-
Machine-Learning</h5>
</ul>
```

Instructions / Execution

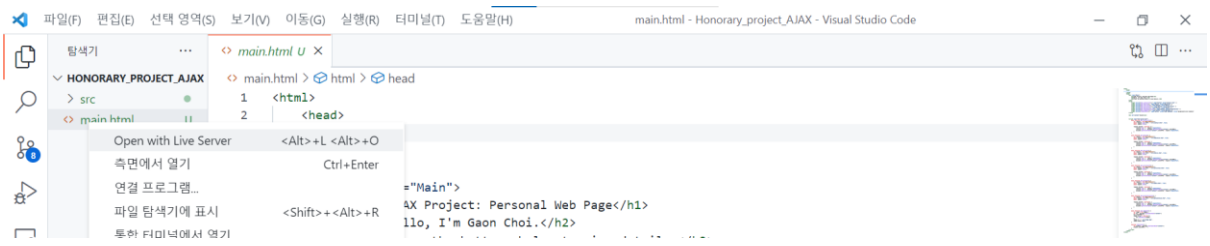
STEP 1

Execution and testing of this project were based on Visual Code. Therefore, execute the Visual Code at first. Bring the entire attached repository in the form of a folder to the application.



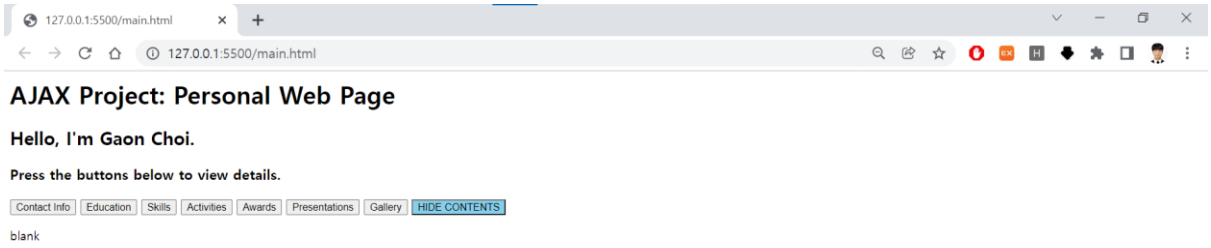
STEP 2

Right click to the “main.html” displayed in the explorer portion. Then, click “Open with Live Server” to run the server to observe the web page implemented on the “main.html”. Then, remind that the invisible form that is running on the Visual Code corresponds to the server, and Chrome is a client.



STEP 3

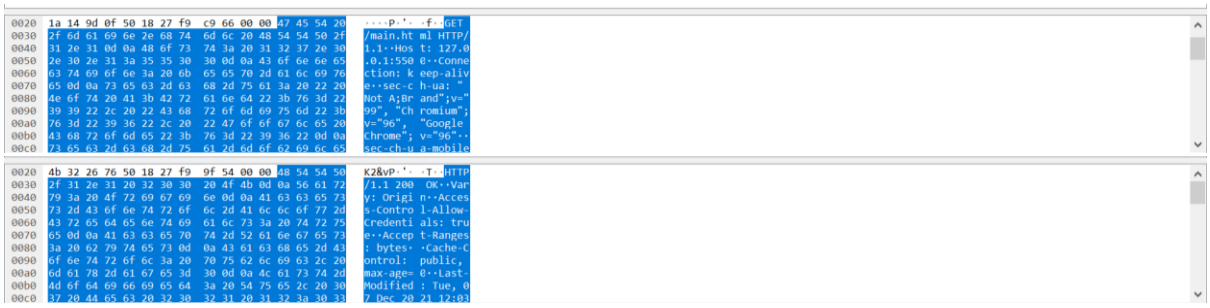
The figure below shows that the server is running with IP address 127.0.0.1 and port number 5500, and the web page in the main.html in that server is successfully displayed.



Results

STEP 1

When the web page was displayed at first, Wireshark gets the following frame.

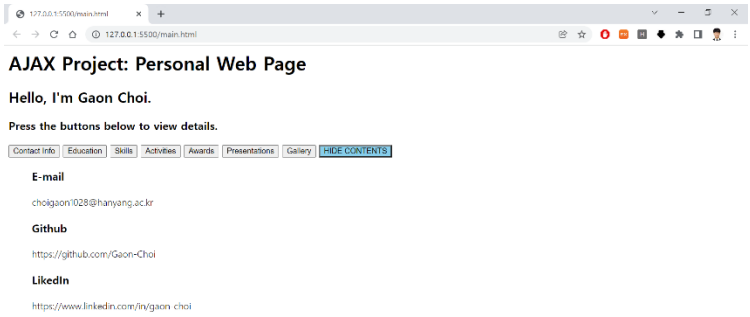


GET /main.html HTTP/1.1

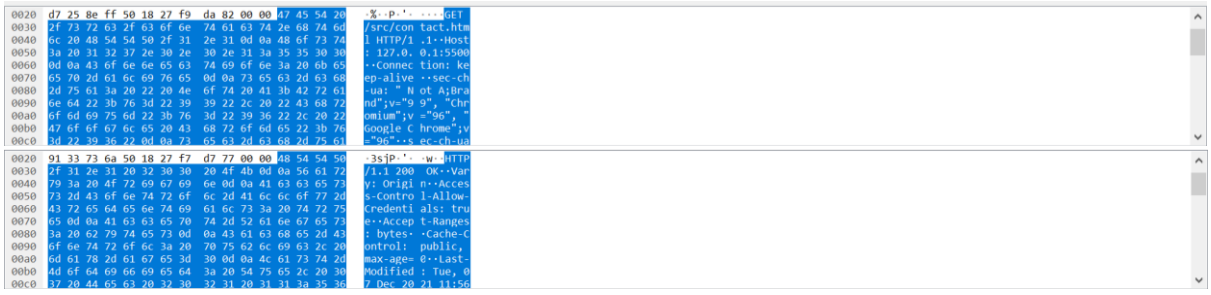
HTTP/1.1 200 OK

STEP 2

Clicking the button "Contact Info", you will get:



Wireshark:

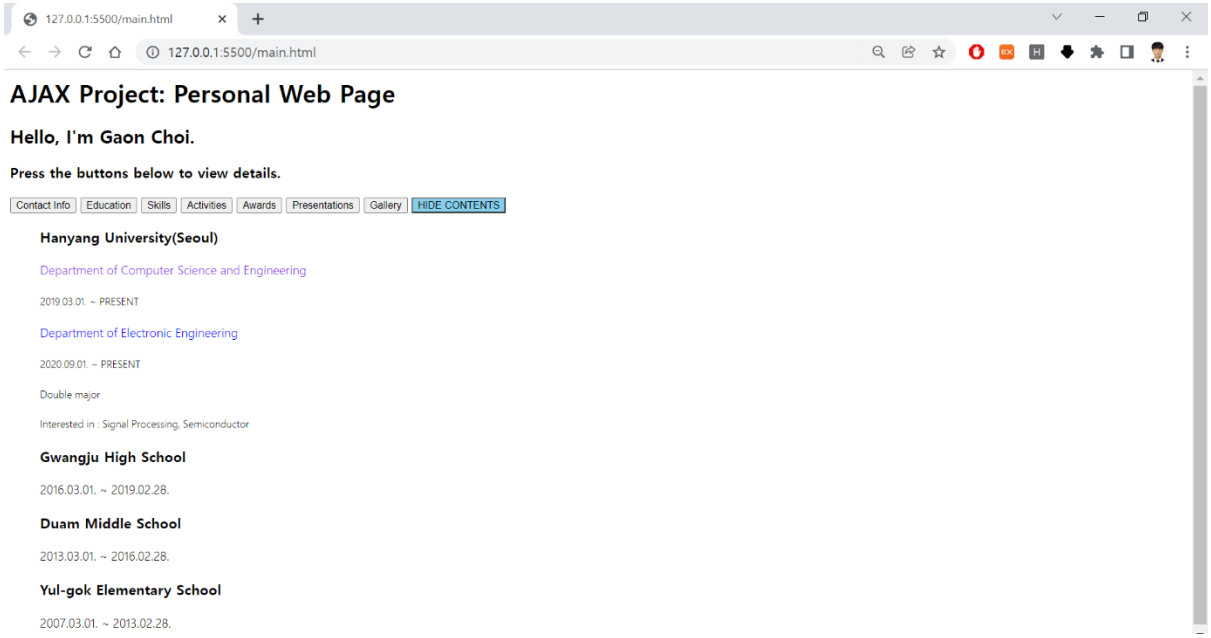


GET /src/contact.html HTTP/1.1

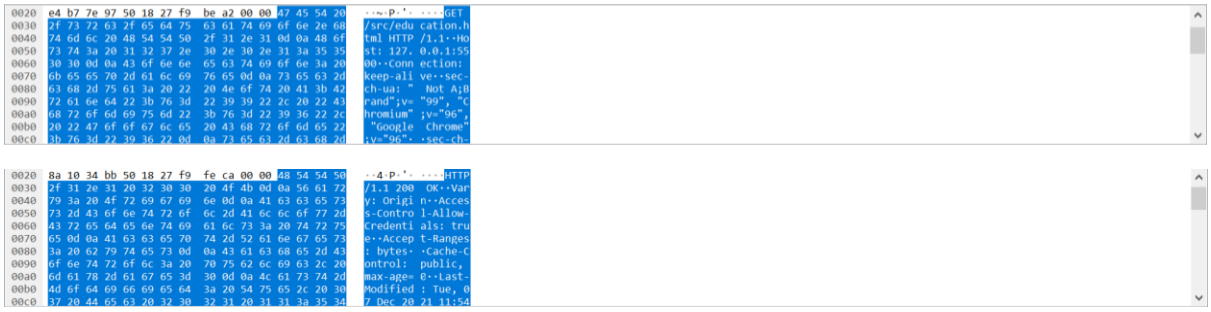
HTTP/1.1 200 OK

STEP 3

Clicking the button “Education”, you will get:

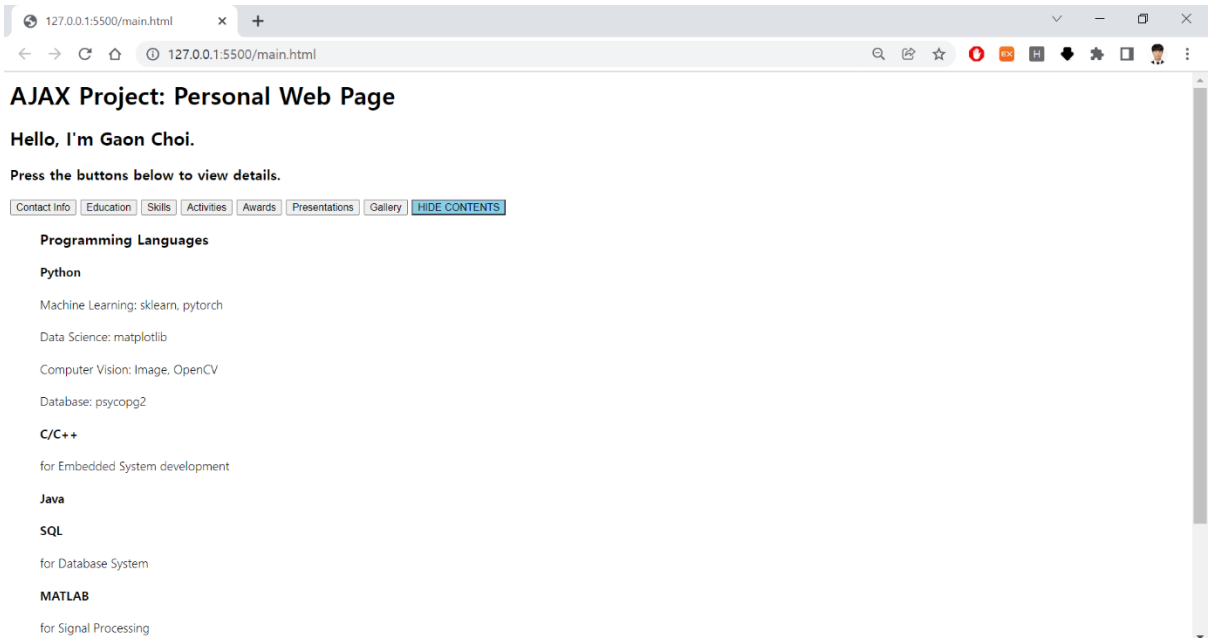


Wireshark:

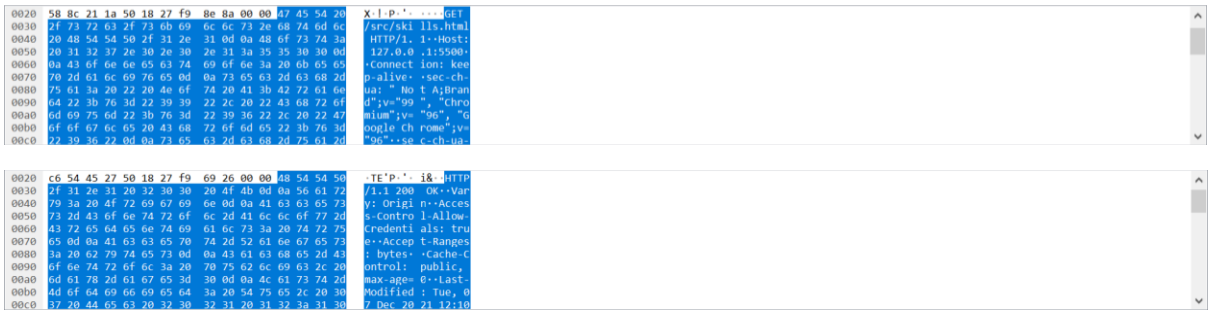


STEP 4

Clicking the button “Skills”, you will get:



Wireshark:

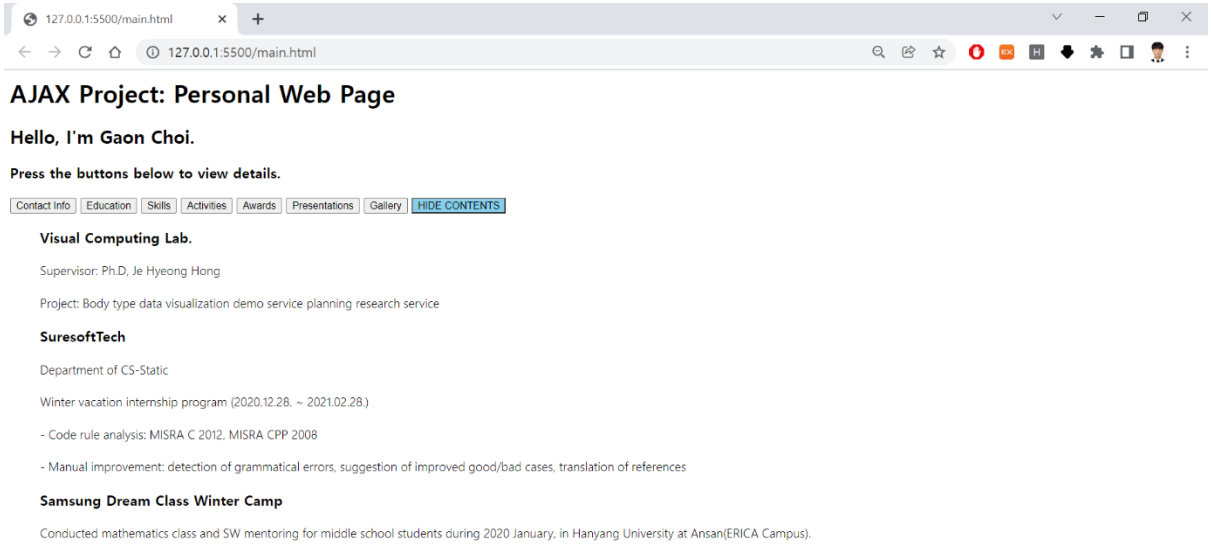


GET /src/skills.html HTTP/1.1

HTTP/1.1 200 OK

STEP 5

Clicking the button “Activities”, you will get:



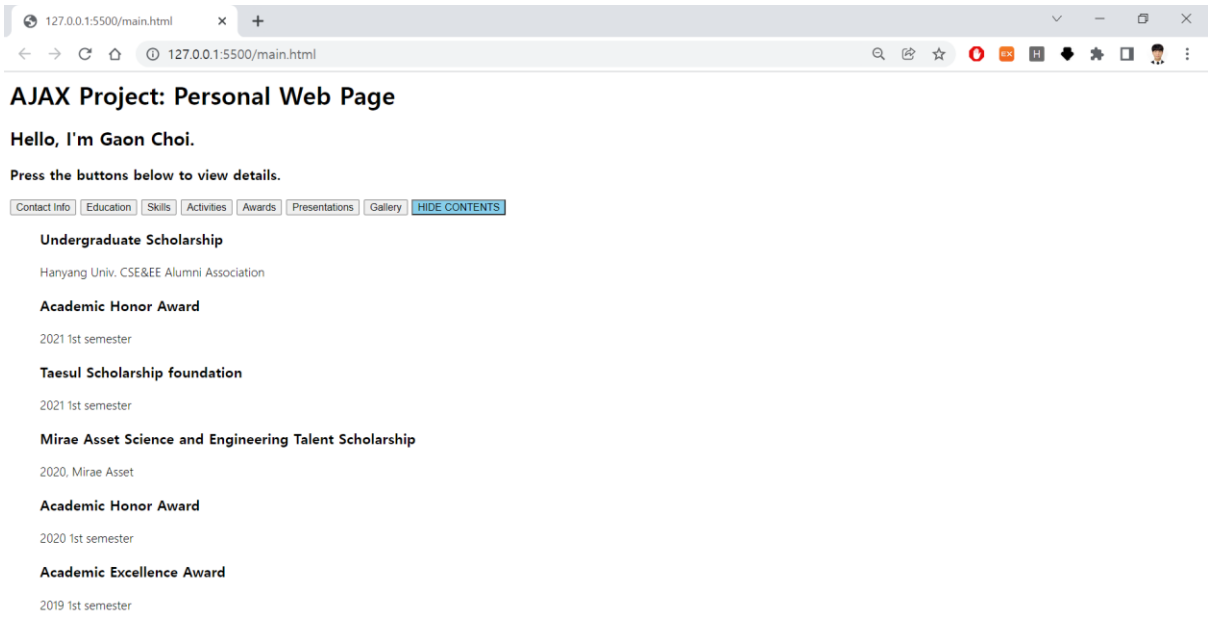
Wireshark:

GET /src/activities.html HTTP/1.1

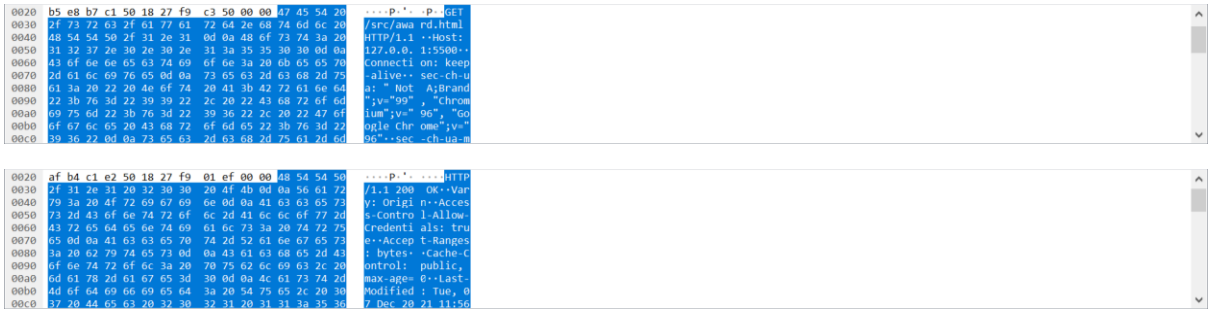
HTTP/1.1 200 OK

STEP 6

Clicking the button “Awards”, you will get:



Wireshark:

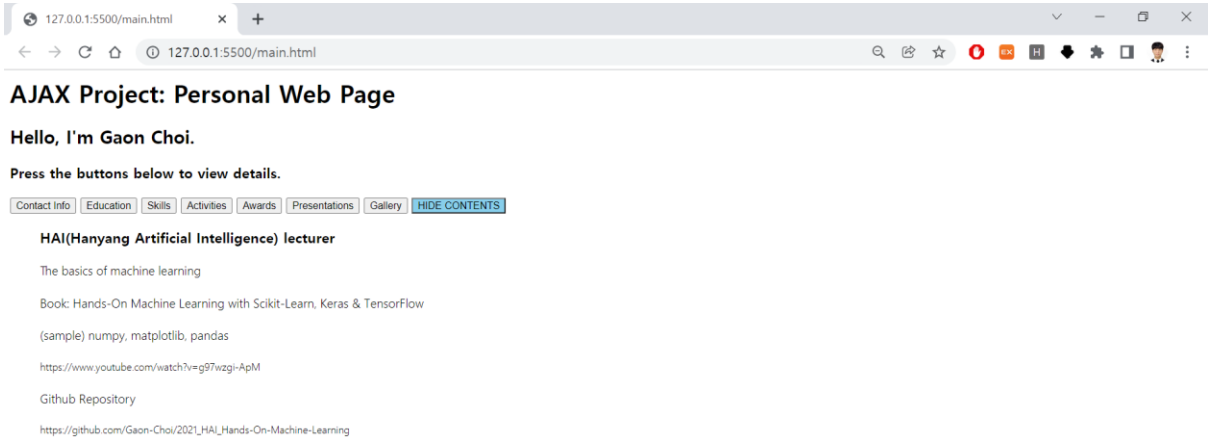


GET /src/award.html HTTP/1.1

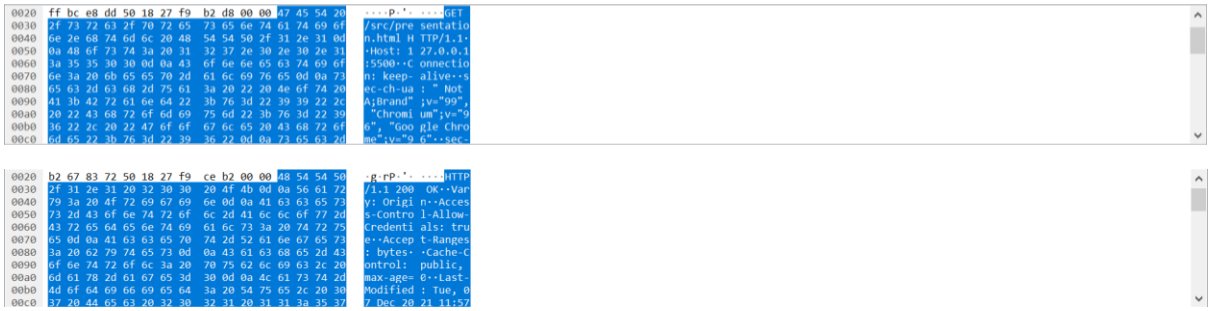
HTTP/1.1 200 OK

STEP 7

Clicking the button “Presentations”, you will get:



Wireshark:

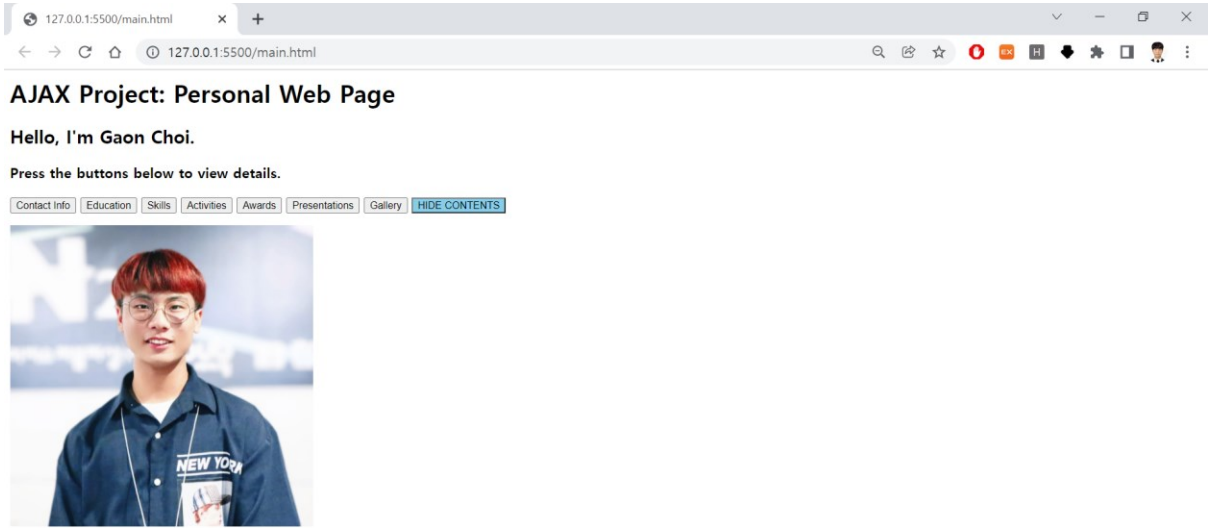


GET /src/presentation.html HTTP/1.1

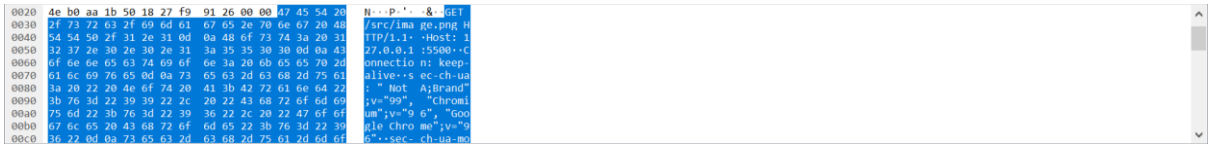
HTTP/1.1 200 OK

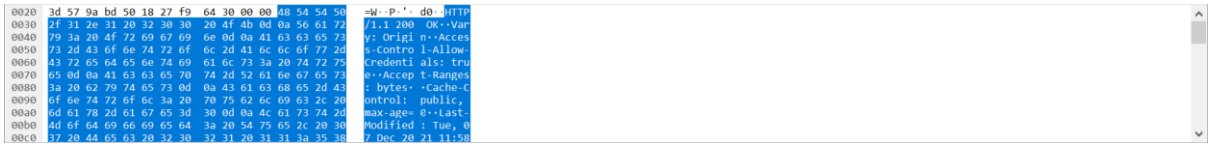
STEP 8

Clicking the button “Gallery”, you will get:



Wireshark:



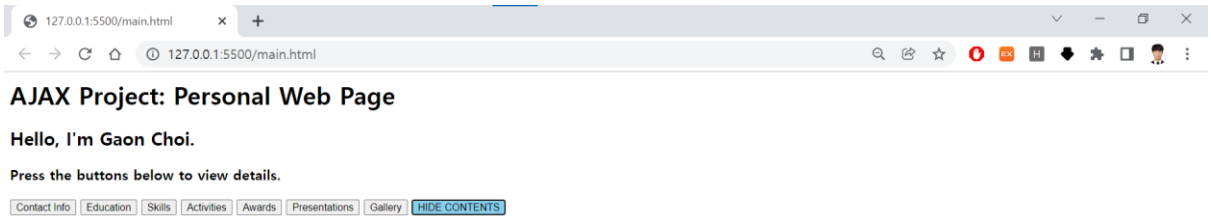


GET /src/image.png HTTP/1.1

HTTP/1.1 200 OK

STEP 9

Clicking the button “HIDE CONTENTS”, you will get:

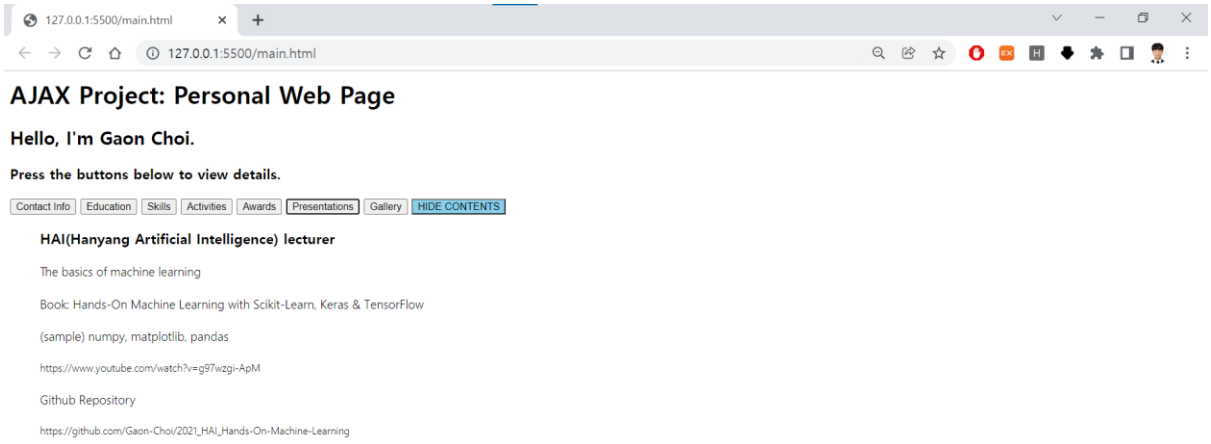


Wireshark:

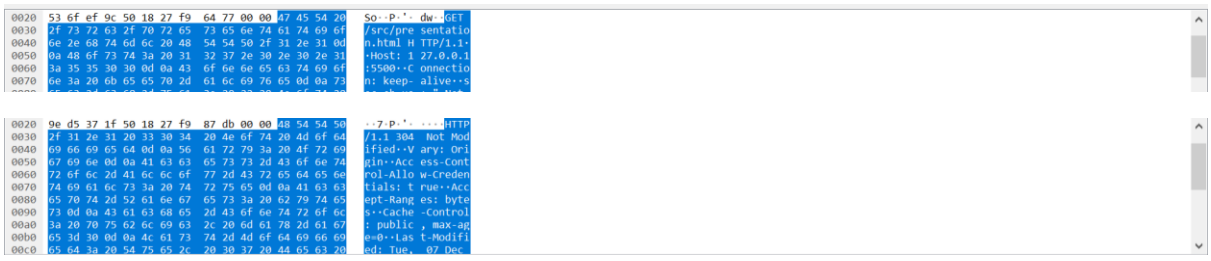
(NOTHING → It just removes “.innerHTML”.

STEP 10

Clicking the button “Presentations” AGAIN, you will get:



Wireshark:



GET /src/presentation.html HTTP/1.1

HTTP/1.1 304 Not Modified

* The client has the traces because it has already loaded once. Therefore, 304 NOT MODIFIED is obtained instead of 200 OK. If “presentation.html” gets modified after the web page is loaded, it will return 200 OK then.

CONCLUSION

Looking at the results from STEP 2 to STEP 10, we can see that when clicking each button, the client(Chrome) sends a GET message to the server to retrieve the HTML file. On the other hand, the GET message for “main.html” has never been called since the web page was initially loaded. It can seen that this doesn’t reload the entire web page, due to the AJAX-based asynchronous processing method, but only some necessary parts (e.g. contact.html, presentation.html, etc.) depending on the situation.

Opinions

When I first started this project, I felt that I was not used to bringing in only a part of the entire web page. However, while understanding the operation and development philosophy of AJAX one by one, I thought that bringing only part of the web page could be a good way to reduce the network overhead as the number of users of the web page increases.

As I proceed with HTML-based development, I feel that I have learned more than simply printing text and pictures one by one in the last project 1. Learning more complex things based on XMLHttpRequest, I think the process of improving one by one, by asking for help from peers or assistants around me, will be a meaningful experience when working as a developer in the industrial field later.