1DA1611:A
Advanced Internet Programming

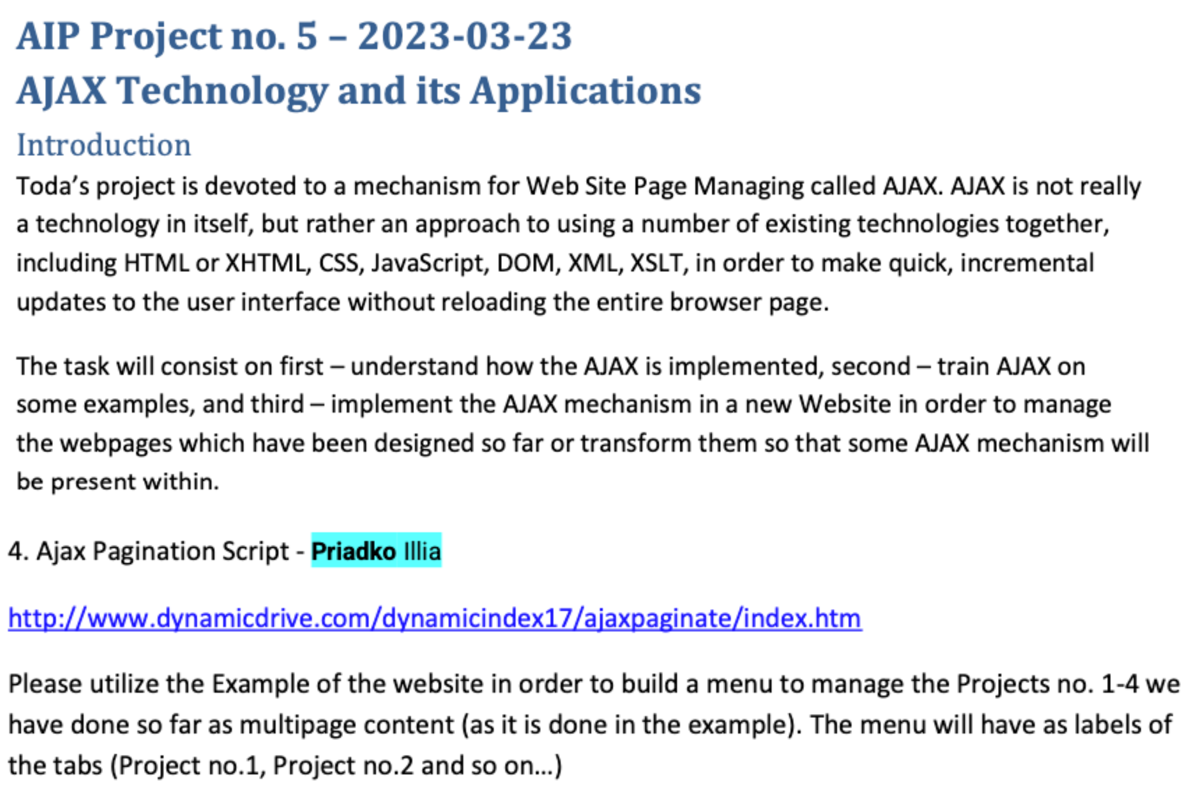
Project nr. 5 Report: 2023-03-23

Illia Priadko (309062)

Plan:

- Combine previous projects in the same root file directory
- Prepare index.html with the page buttons and main content div
- Test and troubleshoot

Given task:



AIP Project no. 5 – 2023-03-23
AJAX Technology and its Applications

Introduction

Toda's project is devoted to a mechanism for Web Site Page Managing called AJAX. AJAX is not really a technology in itself, but rather an approach to using a number of existing technologies together, including HTML or XHTML, CSS, JavaScript, DOM, XML, XSLT, in order to make quick, incremental updates to the user interface without reloading the entire browser page.

The task will consist on first – understand how the AJAX is implemented, second – train AJAX on some examples, and third – implement the AJAX mechanism in a new Website in order to manage the webpages which have been designed so far or transform them so that some AJAX mechanism will be present within.

4. Ajax Pagination Script - **Priadko** Illia

http://www.dynamicdrive.com/dynamicindex17/ajaxpaginate/index.htm

Please utilize the Example of the website in order to build a menu to manage the Projects no. 1-4 we have done so far as multipage content (as it is done in the example). The menu will have as labels of the tabs (Project no.1, Project no.2 and so on…)

Introduction to AJAX:

AJAX (Asynchronous JavaScript and XML) is a technique for creating dynamic and interactive web applications by exchanging data with a web server in the background without requiring a page refresh. It allows web pages to update content dynamically, making the user experience more seamless and faster.

AJAX is based on a combination of technologies including HTML, CSS, JavaScript, and XML/JSON. By leveraging the XMLHttpRequest (XHR) object in JavaScript (as in the example for this report), AJAX allows web applications to send and receive data from a server asynchronously, without interfering with the user's interaction with the web page.

Some use cases where AJAX can be employed include:

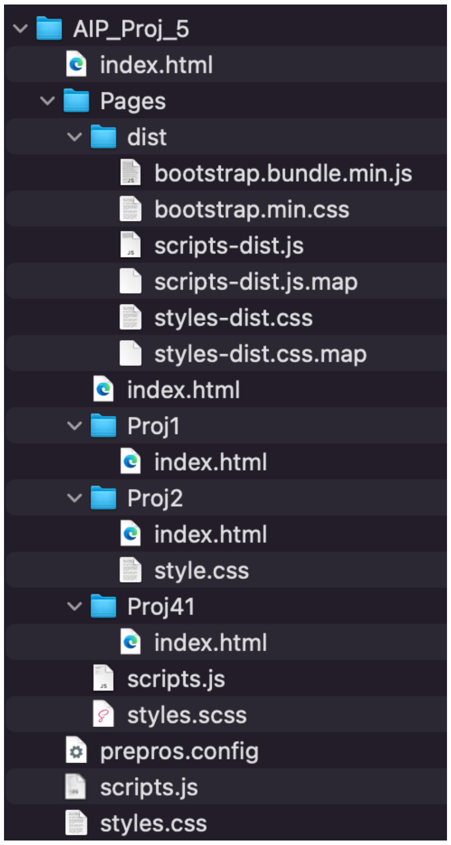
- Implementing dynamic search functionality on a website without requiring a page refresh

- Loading additional content or data into a page dynamically, as the user scrolls down
- Implementing real-time collaboration or chat features in web applications
- Updating parts of a web page after a user interaction, such as a form submission or button click on a menu or paginator.

Overall, AJAX provides a powerful way to create fast, responsive, and interactive web applications that can improve the user experience.

1. **Combine previous projects in the same root file directory**

The structure is as follows:



```
AIP_Proj_5
    index.html
    Pages
        dist
            bootstrap.bundle.min.js
            bootstrap.min.css
            scripts-dist.js
            scripts-dist.js.map
            styles-dist.css
            styles-dist.css.map
        index.html
        Proj1
            index.html
        Proj2
            index.html
            style.css
        Proj41
            index.html
        scripts.js
        styles.scss
    prepros.config
    scripts.js
    styles.css
```

The main root directory contains the main index.html with the paginator itself, the /Pages/ contains the project 4, which in turn already contains the rest of projects in their respective folders.

2. **Prepare index.html with the page buttons and main content div**

The page is designed to have two rows of buttons in the header and footer, to match the example. Since only 4 pages will be there, the first and last page buttons were not implemented, due to all of the buttons being visible on-screen at all times.

Between those header and footer elements, there will be a <div> element that will be populated with the content of the selected page. In case nothing is selected yet, a prompt will show a request to select one of the pages.

The HTML code is as follows:

```html
<!doctype html>
<html lang="en">

<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">

    <!-- CSS -->
    <link rel="stylesheet" href="./Pages/dist/bootstrap.min.css">
    <link rel="stylesheet" href="./Pages/dist/styles-dist.css">
    <link rel="stylesheet" href="./styles.css">
    <title>Project 5: Page-view of projects 1-4</title>
</head>

<body>
    <!-- Header -->
    <header>
        <div class="navbar navbar-dark bg-dark shadow-sm">
            <div class="container">
                <div class="btn-group" role="group">
                    <button type="button" class="btn btn-primary" onclick="loadDoc(1)">Project
1</button>
                    <button type="button" class="btn btn-primary" onclick="loadDoc(2)">Project
2</button>
                    <button type="button" class="btn btn-primary" onclick="loadDoc(3)">Project
3</button>
                    <button type="button" class="btn btn-primary" onclick="loadDoc(4)">Project
4</button>
                </div>
            </div>
        </div>
    </header>

    <!-- Main container -->
    <main>
        <div class="container my-main">
            <div id="content">
                <br><br>
                <h2>Please choose a project in the header to view.</h2>
                <br><br>
            </div>
        </div>
    </main>

    <!-- Footer -->
    <footer class="text-muted bg-dark py-5 my-footer">
        <div class="container">
            <div class="btn-group" role="group">
                <button type="button" class="btn btn-primary" onclick="loadDoc(1)">Project 1</button>
                <button type="button" class="btn btn-primary" onclick="loadDoc(2)">Project 2</button>
                <button type="button" class="btn btn-primary" onclick="loadDoc(3)">Project 3</button>
                <button type="button" class="btn btn-primary" onclick="loadDoc(4)">Project 4</button>
            </div>
        </div>
    </footer>
    <!-- Load required scripts -->
    <script src="./Pages/dist/bootstrap.bundle.min.js"> </script>
    <script src="./scripts.js"> </script>
    <script src="./Pages/dist/scripts-dist.js"></script>
</body>

</html>
```

To run the paginator, JavaScript must be implemented.
In the HTML file, the loadDoc function is called with a specific argument depending on the button pressed. The if-statements then decide which page to load and show inside the content div.

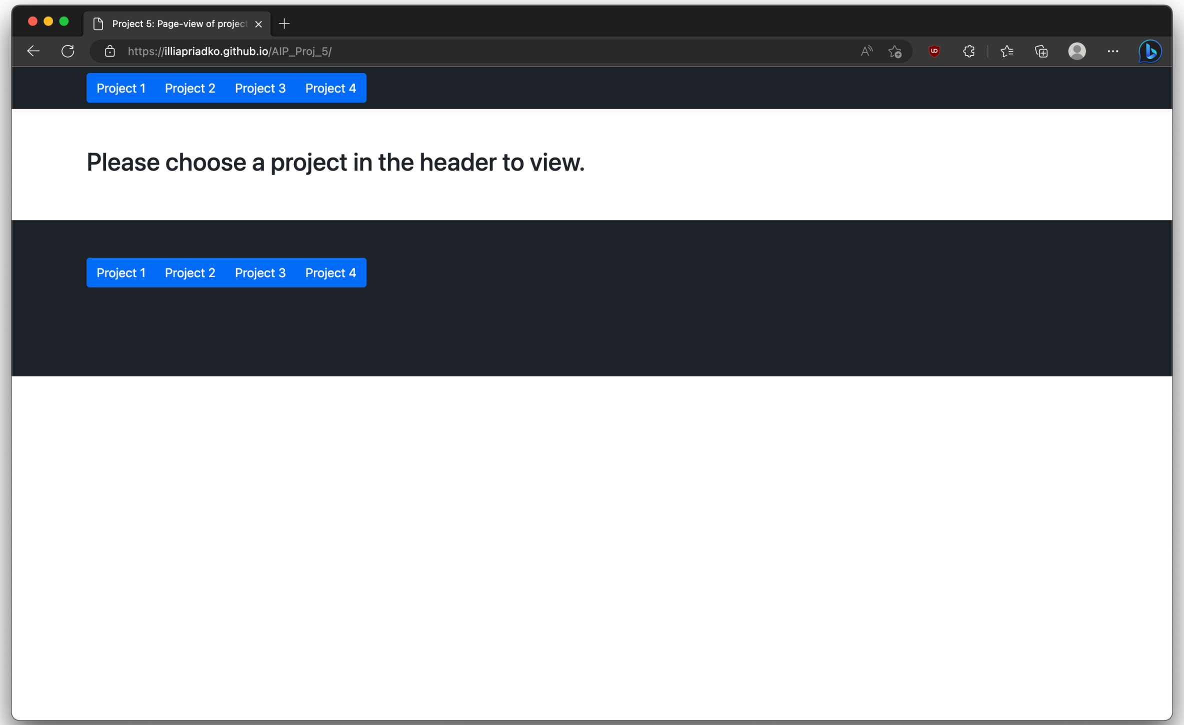
```javascript
const httpRequest = new XMLHttpRequest();

function loadDoc(page) {
    httpRequest.onload = function () {
        document.getElementById("content").innerHTML = this.responseText;
    }
    if (page == 1) {
        httpRequest.open("GET", "./Pages/Proj1/index.html", true);
        httpRequest.send();
    }
    if (page == 2) {
        httpRequest.open("GET", "./Pages/Proj2/index.html", true);
        httpRequest.send();
    }
    if (page == 3) {
        httpRequest.open("GET", "./Pages/Proj41/index.html", true);
        httpRequest.send();
    }
    if (page == 4) {
        httpRequest.open("GET", "./Pages/index.html", true);
        httpRequest.send();
    }
}
```

Another step that was necessary, was to import scripts from those subpages into the scripts for index.html. Due to the fact AJAX was not able to import scripts into the correct section, this was a necessary step.

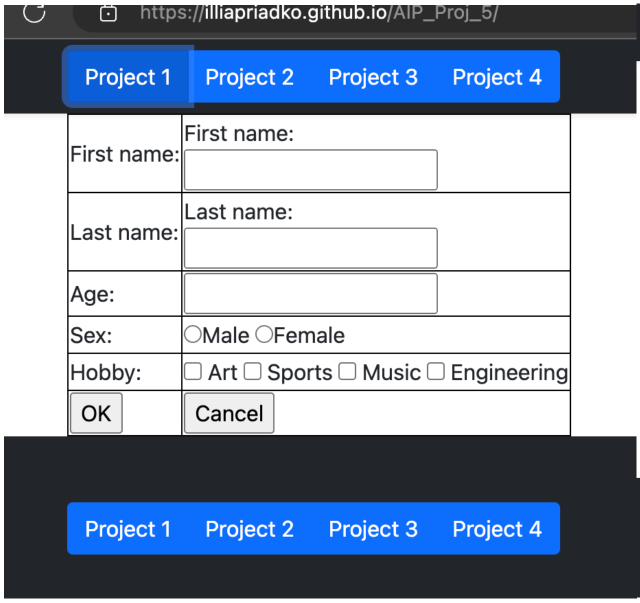
3. Test and troubleshoot

The page was tested via Prepros (acting as a local webserver), and looks like this:


Project 5: Page-view of project
https://illiapriadko.github.io/AIP_Proj_5/
Project 1
Project 2
Project 3
Project 4

Please choose a project in the header to view.

And when clicking the page buttons:

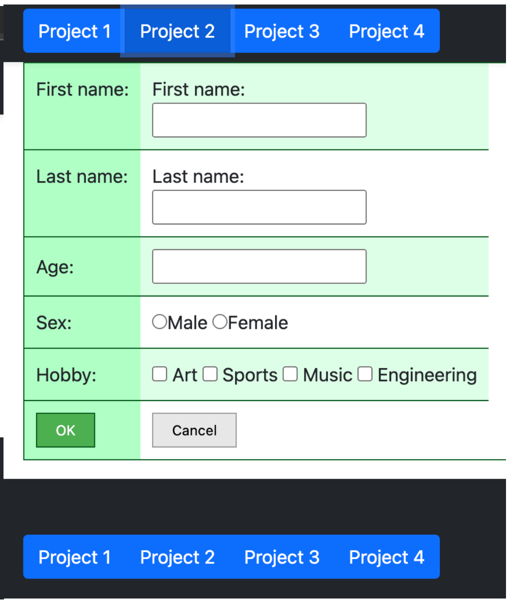


https://illiapriadko.github.io/AIP_Proj_5/

Project 1 Project 2 Project 3 Project 4

First name:	First name:
Last name:	Last name:
Age:	
Sex:	◯Male ◯Female
Hobby:	☐ Art ☐ Sports ☐ Music ☐ Engineering
OK	Cancel

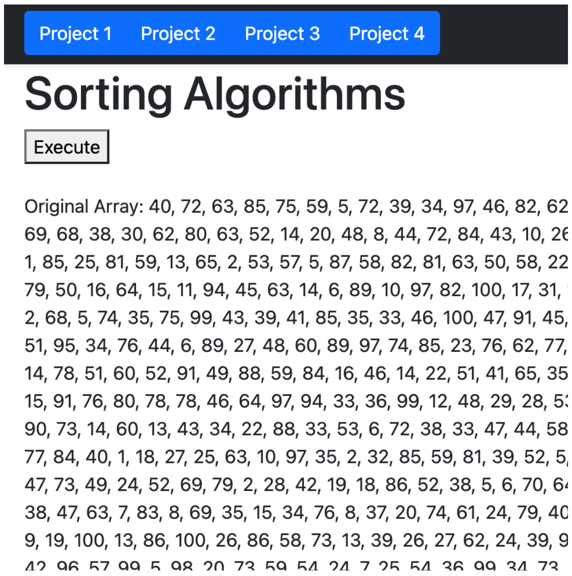
Project 1 Project 2 Project 3 Project 4



Project 1 Project 2 Project 3 Project 4

First name:	First name:
Last name:	Last name:
Age:	
Sex:	◯Male ◯Female
Hobby:	☐ Art ☐ Sports ☐ Music ☐ Engineering
OK	Cancel

Project 1 Project 2 Project 3 Project 4

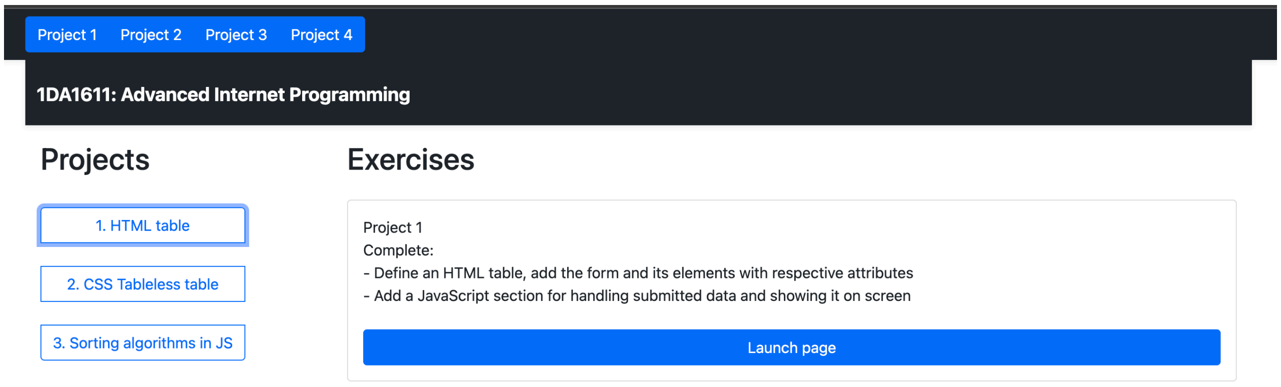


Project 1 Project 2 Project 3 Project 4

Sorting Algorithms

Execute

```
Original Array: 40, 72, 63, 85, 75, 59, 5, 72, 39, 34, 97, 46, 82, 62
69, 68, 38, 30, 62, 80, 63, 52, 14, 20, 48, 8, 44, 72, 84, 43, 10, 2
1, 85, 25, 81, 59, 13, 65, 2, 53, 57, 5, 87, 58, 82, 81, 63, 50, 58, 22
79, 50, 16, 64, 15, 11, 94, 45, 63, 14, 6, 89, 10, 97, 82, 100, 17, 31,
2, 68, 5, 74, 35, 75, 99, 43, 39, 41, 85, 35, 33, 46, 100, 47, 91, 45,
51, 95, 34, 76, 44, 6, 89, 27, 48, 60, 89, 97, 74, 85, 23, 76, 62, 77,
14, 78, 51, 60, 52, 91, 49, 88, 59, 84, 16, 46, 14, 22, 51, 41, 65, 35
15, 91, 76, 80, 78, 78, 46, 64, 97, 94, 33, 36, 99, 12, 48, 29, 28, 5
90, 73, 14, 60, 13, 43, 34, 22, 88, 33, 53, 6, 72, 38, 33, 47, 44, 58
77, 84, 40, 1, 18, 27, 25, 63, 10, 97, 35, 2, 32, 85, 59, 81, 39, 52, 5,
47, 73, 49, 24, 52, 69, 79, 2, 28, 42, 19, 18, 86, 52, 38, 5, 6, 70, 6
38, 47, 63, 7, 83, 8, 69, 35, 15, 34, 76, 8, 37, 20, 74, 61, 24, 79, 40
9, 19, 100, 13, 86, 100, 26, 86, 58, 73, 13, 39, 26, 27, 62, 24, 39, 9
42, 96, 57, 99, 5, 98, 20, 73, 59, 54, 24, 7, 25, 54, 36, 99, 34, 73
```



Project 1 Project 2 Project 3 Project 4

1DA1611: Advanced Internet Programming

Projects

1. HTML table

2. CSS Tableless table

3. Sorting algorithms in JS

Exercises

Project 1
Complete:
- Define an HTML table, add the form and its elements with respective attributes
- Add a JavaScript section for handling submitted data and showing it on screen

Launch page

Scripts, including the subpages' scripts are loaded and working.

Some troubleshooting was needed after I have discovered the issue with loading subpage scripts, as well as inline css from project 3 causes the font to change on the whole page. This was fixed by removing inline css .body element with a font.

The page, along with the rest of projects done so far, are deployed via git, to my personal GitHub repository for easy access: https://illiapriadko.github.io/AIP_Proj_5/

Conclusion: After having completed this project, I had learned about AJAX, how it is used, was able to implement it in my own page, as well as learned about some hidden issues that may arise when doing so.