1DA1611:A
Advanced Internet Programming

Project nr. 12 Report: 2023-05-18

Illia Priadko (309062)

Plan:

- Add the rest of projects to the list, as well as repository
- Revamp the looks, add a more functional navbar
- Add PDF download button and task description to sit above main content
- Add tooltips for each project's button

Add all projects to the list

The project is rebuilt from scratch, using Bootstrap 5 components, similar to the previous ones. It also uses a similar AngularJS component to fetch all project info from a JSON file, making the list easier to edit and maintain.

Here is the code snippet for the AngularJS module:

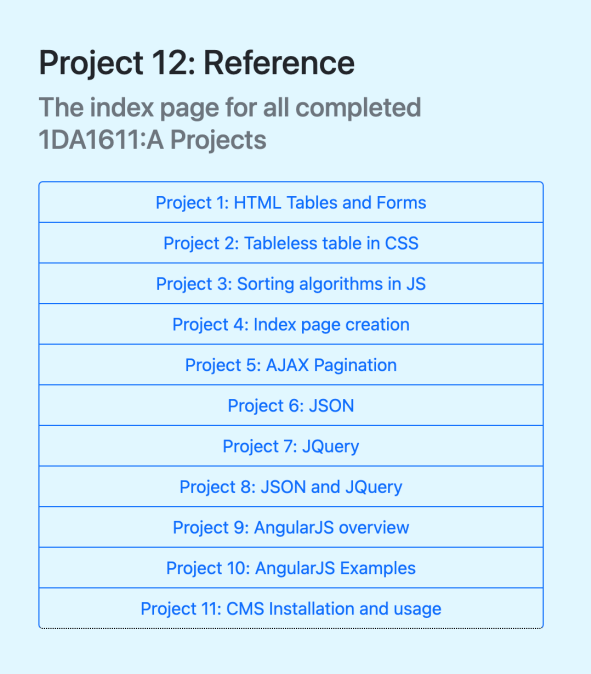
```
    <script>
        const httpRequest = new XMLHttpRequest();
        // my application to populate buttons
        var app = angular.module('myApp', []);
        app.controller('myController', function ($scope, $http) {
            // fetch .json from directory
            $http.get('./projects.json').then(function (response) {
                $scope.buttons = response.data;
            });
            $scope.buttonClicked = function (button) {
                // Handle button click event
                loadDoc(button.id, button.desc);
            };
        });


        //loadDoc function
        function loadDoc(page, desc) {
            var contentDiv = document.getElementById("main-content");
            var iframe = document.createElement("iframe");
            iframe.src = page; // Replace with the desired page URL
            iframe.width = "100%";
            iframe.height = "900";
            contentDiv.innerHTML = ""; // Clear existing content
            contentDiv.appendChild(iframe); // Append the iframe to the content div
            document.getElementById("pdf-download").setAttribute("href", page +
"/report.pdf");
            document.getElementById("description").innerHTML = desc;
        }
```

```
    </script>
```

Here one can see the script fetching project info from the .json file and feeding it into the loadDoc function. loadDoc was remade to now load html files using an <iframe> element, as that is a much better approach for this purpose, not breaking scripts and styles.
The json is formatted in the following way:

```
[
    {
        "id": "./Pages/Proj1",
        "label": "Project 1",
        "title": "HTML Tables and Forms",
        "desc": "After having completed this project, I got familiar with HTML forms,
JavaScript basic syntax (variable declaration, if-statements and loops), was able to
successfully use its functions for getting information from HTML elements, and
gathering it in one string for output. I have also familiarised myself with
development tools provided inside Chromium."
    },
```

In the end, the section with buttons looks like this:

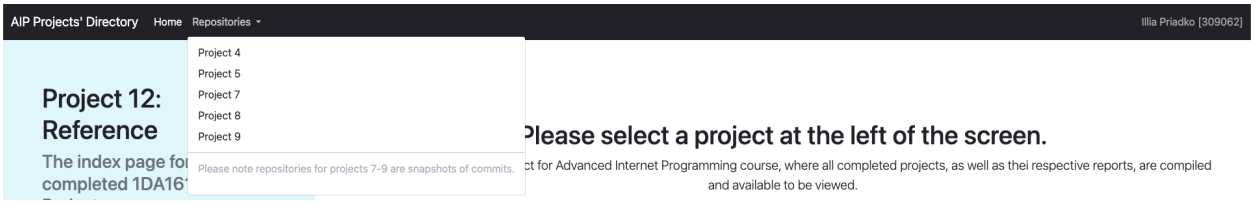


Navbar creation

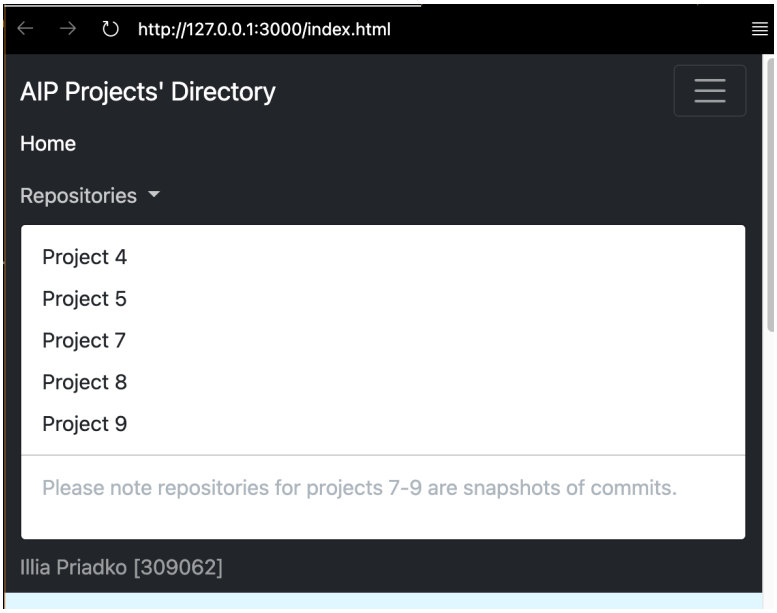
Navbar was a pretty straight-forward component to implement, and to make it easier, I decided it would be a good place to put the home link, the drop-down list of repositories

related to AIP, and my name. What is also a nice feature of Bootstrap is the column system: meaning the site is responsive.

Desktop view:

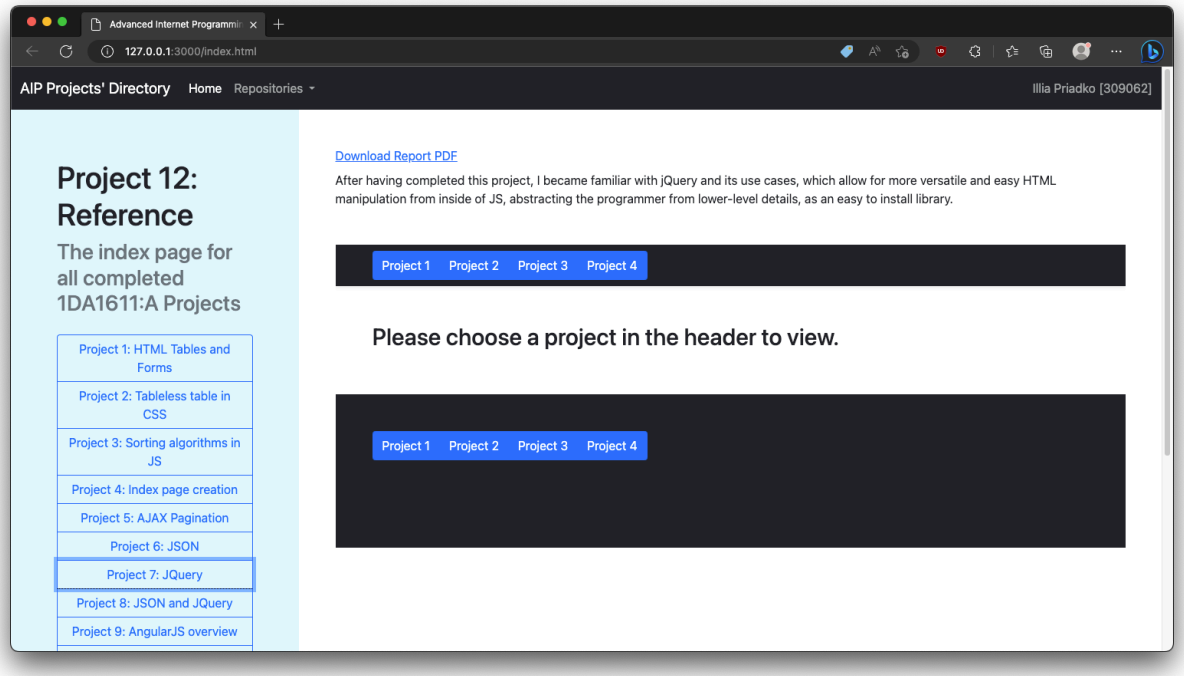


Mobile view:



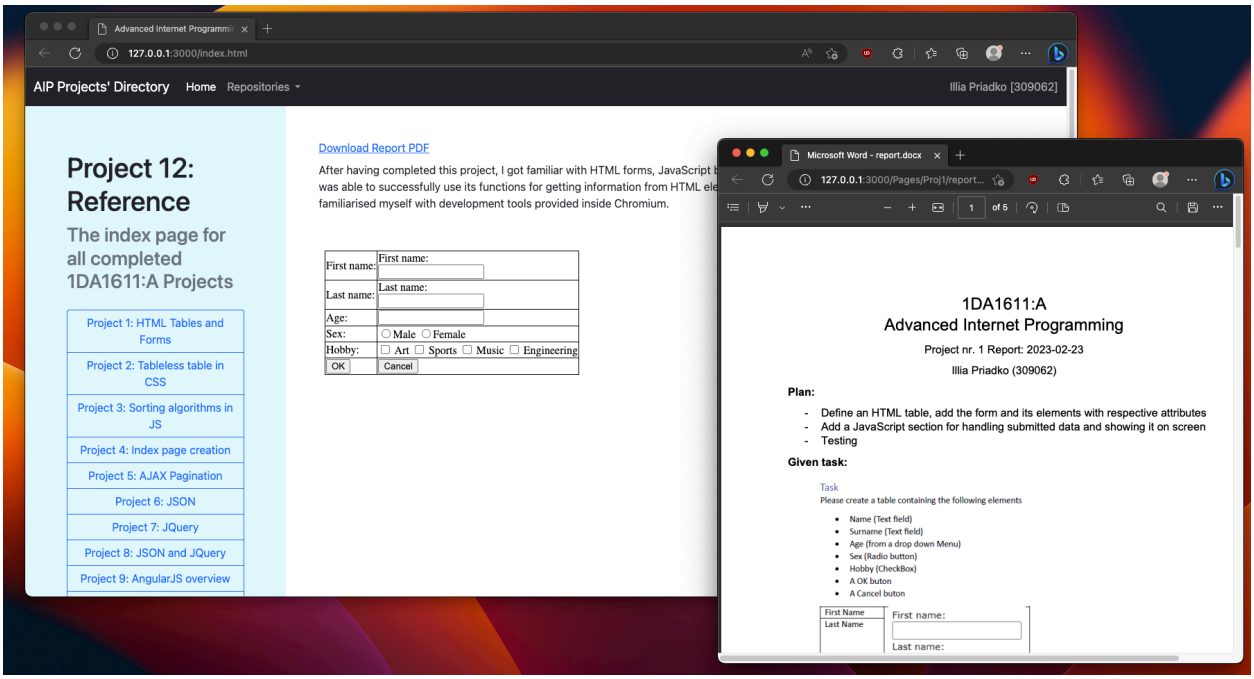
Report section:

As the buttons are clicked, they call a function to populate the right (or bottom) part of the screen with the content itself. A good feature of this approach is that the <iframe> allows to retain all scripts and styles from the page we are appending, a much better approach than using AJAX for this particular application.



Another two required features of the website are report download links, and descriptions for each project. Well, as mentioned previously, description would be stored in the “desc” value for each position in the “projects.json” file fetched and loaded with AngularJS.
Howerer, one also has to figure out how to populate the report link. An option of making a new folder called “reports” was considered, but ultimate one would need to make another “path” key for each project’s json entry. In the end, I decided to actually reuse the same “path” that points to the .html file of the project itself, but to add “./report.pdf” to it, for the <a> link to send the user to the report page, where they can view it directly in browser. The report, as expected, is located in the same directory as the page itself.

```
            contentDiv.appendChild(iframe); // Append the iframe to the content div
            document.getElementById("pdf-download").setAttribute("href", page + "/report.pdf");
            document.getElementById("description").innerHTML = desc;
```



Adding tooltips for each project's button:

Despite Bootstrap coming often installed in a bundle together with Popper – a framework for implementing pop-overs, it was not applicable in this case. This is due to the fact the buttons are actually not pre-defined from the start – they are generated by AngularJS application upon loading of the page and executing the included scripts. Hence, I had to come up with another solution for this problem.

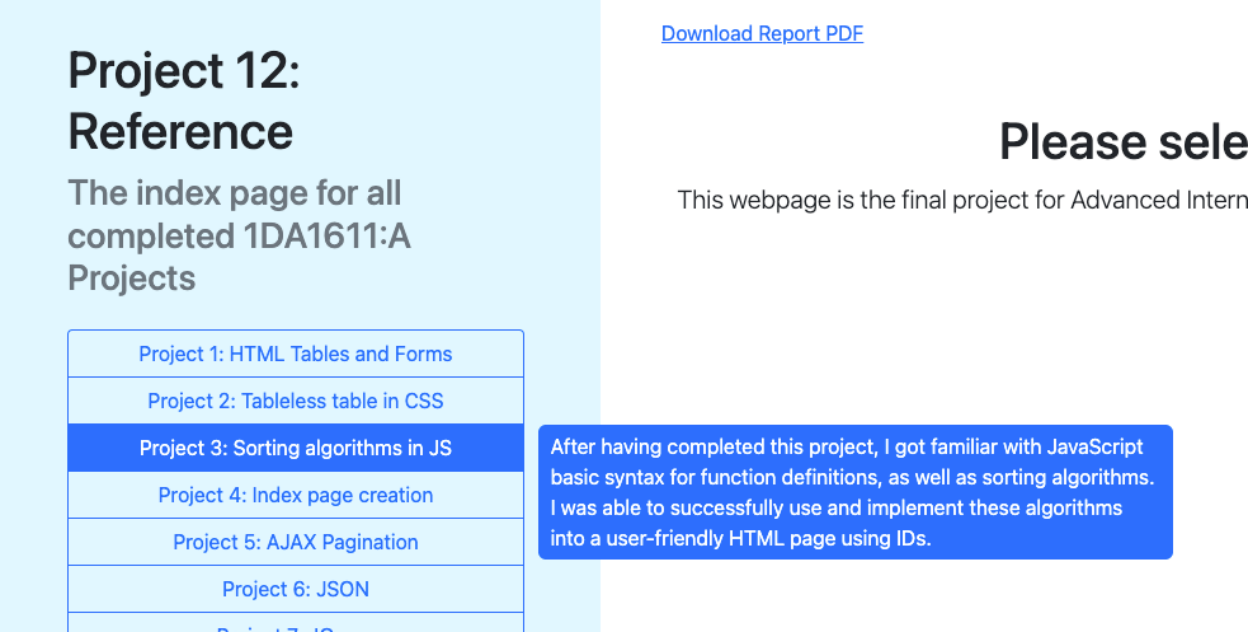
```
<button type="button" ng-repeat="button in buttons" ng-click="buttonClicked(button)"
        class="btn btn-outline-primary deschover">
        <!-- Expression with button name and date -->
        {{button.label+": "+button.title}}<span
        class="deschovertext deschover-right">{{button.desc}}</span>
</button>
```

In the code snippet above, a small change is noticeable: a new class for the button template, which will be filled and repeated based on the .json, called “deschover” (description hover), and a element that is responsible for slowing the tooltip itself. Both definitions for these classes are in the stylesheet for this page, made to match the overall look of the site and the colour theme of Bootstrap.

```
/* deschover container */
.deschover {
    position: relative;
    display: inline-block;
    border-bottom: 1px dotted black;
    /* If you want dots under the hoverable text */
}

/* deschover text */
.deschover .deschovertext {
    visibility: hidden;
    width: 500px;
```

```
    background-color: #0d6efd;
    color: #ffffff;
    text-align: left;
    padding: 5px 10px;
    border-radius: 6px;

    /* Position the deschover text */
    position: absolute;
    z-index: 1;
    top: 0%;
    left: 120%;
    margin-left: -60px;

    /* Fade in deschover */
    opacity: 0;
    transition: opacity 0.3s;
}

/* Show the deschover text when you mouse over the deschover container */
.deschover:hover .deschovertext {
    visibility: visible;
    opacity: 1;
}
```

The contents of the tooltip are also, in fact, fetched from the .json file, and filled with the same description as the area above main content. The user, in the end, can preview the description of the project while navigating the page and make sure they choose the right one.



The final page is available on GitHub Pages:
[Advanced Internet Programming Project 12 (illiapriadko.github.io)](illiapriadko.github.io)

Conclusions:
After having made this project, I was able to use many of the skills and knowledge I have obtained during the course of Advanced Internet Programming, especially JavaScript, AngularJS, and CSS styling. It all blends well together, and I am now more

confident in using the different frameworks and skills to use in my future career, whenever anything Front-End related would come up as a task.