PRACTICE 3-4

DEVELOPMENT OF THE WEB - SITE, USING JAVA SCRIPT

JAVASCRIPT BASICS

JavaScript is a programming language that adds interactivity to your website. This happens in games, in the behavior of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc.

JavaScript itself is relatively compact, yet very flexible.

- Browser Application Programming Interfaces (APIs) built into web browsers, providing functionality such as dynamically creating HTML and setting CSS styles; collecting and manipulating a video stream from a user's webcam, or generating 3D graphics and audio samples.
- Third-party APIs that allow developers to incorporate functionality in sites from other content providers, such as Twitter or Facebook.
- Third-party frameworks and libraries that you can apply to HTML to accelerate the work of building sites and applications.

Go to your test site and create a new folder named scripts. Within the scripts folder, create a new text document called **main.js**, and save it.

Creating the JS file:

In your **index.html** file, enter this code on a new line, just before the closing **</body>** tag:

```
<script src="scripts/main.js"></script>
```

This is doing the same effect as the <link> element for CSS. It applies the JavaScript to the page, so it can have an effect on the HTML (along with the CSS, and anything else on the page).

Add this code to the **main.js** file:

```
const myHeading = document.querySelector('h1');
myHeading.textContent = 'Hello, my name is Alla Jammine!';
```

Make sure the HTML and JavaScript files are saved. Then load index.html in your browser.

WHAT CAN JAVASCRIPT DO IN A BROWSER?

JavaScript is a "safe" programming language. It does not provide low-level access to memory or processor, because it was originally created for browsers that do not require it.

The capabilities of JavaScript are highly dependent on the environment in which it works. For example, Node.JS supports the functions of reading/writing arbitrary files, executing network requests, etc.

JavaScript is available in the browser for everything related to manipulating web pages, interacting with the user and the web server.

For example, in a JavaScript browser, you can:

- Add new HTML code to the page, modify existing content, modify styles.
- React to user actions, mouse clicks, pointer movements, keystrokes.
- Send network requests to remote servers, download and upload files (AJAX and COMET technologies).
- Receive and set cookies, ask questions to the visitor, show messages.
- Store data on the client side ("local storage").

WHAT CAN JAVASCRIPT NOT DO IN THE BROWSER?

JavaScript capabilities in the browser are limited for the sake of user safety. The purpose is to prevent an unscrupulous web page from accessing personal information or damaging user data.

Examples of such restrictions include:

- JavaScript on a web page cannot read/write arbitrary files on the hard disk, copy them, or run programs. It does not have direct access to the OS system functions.

Modern browsers allow it to work with files, but with limited access, and provide it only if the user performs certain actions, such as "dragging" a file into the browser window or selecting it using the <input> tag.
There are ways to interact with the camera/microphone and other devices, but they require explicit user permission. Thus, a JavaScript-enabled page cannot imperceptibly turn on a webcam, observe what is happening and send information to the FSB.

- The different windows/tabs are unaware of each other. Sometimes one window, using JavaScript, opens another window. But even in this case, JavaScript from one page does not have access to another if they came from different sites (from a different domain, protocol or port).

This is called the "Same Origin Policy". To circumvent this restriction, both pages must agree to this and contain JavaScript code that exchanges data in a special way.
This restriction is necessary, again, for the safety of the user. Page https://anysite.com the one that the user opened should not have access to another browser tab with the URL https://gmail.com and steal information from there.

- JavaScript can easily interact with the server from which the current page came. But its ability to receive data from other sites/domains is limited. Although this is possible in principle, which requires explicit consent (expressed in HTTP headers) with the remote party. Again, this is a security limitation.

INDIVIDUAL TASK:

1. Create your own **Movie Page** site including movie list (for Example **Netflix/Disney**) using HTML, CSS and JAVA SCRIPT.
2. Make a folder of movie images.
3. Create a 3 type of files (index.html; css; app.js) in Code Visual studio.
4. For main look use **google font / fontawesome.com**
5. Use **Nav_bar/ Side_bar** for your site
6. Create your profile there with the **toggle – icon** and **toggle ball** next to it
7. Create a Movie Slider Using JS, back ground color.
8. Make slider Responsive (after the scrolling last movie it should go back) using JS (limits, InnerWidth, Click Counter, Ratio or time setup).