Assignment 1

In this assignment we will cover the basics of web servers and clients by making our custom web server to perform the task at hand.

Prerequisites:

* Install Node.js from <https://nodejs.org/>. You can use either LTS or the latest version.
* Install Express web framework using NPM: <https://expressjs.com/>.
* Install an EJS template engine via NPM: <https://ejs.co/>

## Task 1

In this task you will create a simple server that checks the age of the visitors and greets them by their name. Your server must take the name of the user from a request URL and user’s age from URL query string as follows:

Request at /John?age=21 greets John by his name

Request at /John?age=2 tells that John is too young to visit the website

Requests at /John/\*?age=2 should redirect the user to /John?age=2 (\* stands for any page)

Request at /John prompts the user to enter their age via an HTML form

Request at / (home page) prompts the user to enter their name and redirects to the corresponding page in JavaScript.

The name was chosen as an example here. Your server must work with any provided name. You may choose any value as the minimum required age.

Hints:

* You can send the page to the user using the res.send() method in Express
* If you choose to send an HTML file, you can use res.sendFile()
* Use the path module of Node when working with file paths
* res.redirect() can be used to redirect users to another page
* Query parameters are available via req.query
* URL parameters are available via req.params
* HTML forms can be used to submit data automatically
* You can redirect a user to a different page by setting the value of window.location
* Node, Express and EJS docs are incredibly helpful

## Task 2

In this task you must improve the application created in task 1. The object is to keep the functionality the same, but actually style your webpage by including some CSS. Your page must be formatted according to the following rules:

* All html in both files should use helvetica font with a 24 pixel size

**The following changes must be made in home.html**

* The text element “Please enter your name to continue” should be center aligned
* The text box element should utilize transitions. All transitions should take 0.5 seconds
  + The text box width should ease in and out: the initial width should be 165 pixels and when the box is focused on it should be expanded to 330px.
  + When there is no text in the text box it should be pink, when a user enters text, the box should transition to white.
* The text box should be 165 pixels x 32 pixels (width x height)
* The text box should have half a character width margin between it and any other element on the same horizontal plane.
* The text box should have rounded corners with a 4 pixel radius
* The submit button should have a padding width of three characters
* The text requirements for the button are 18 pixel Helvetica
* The margins should be set such that the bottom of the button aligns with the bottom of the text box. Additionally there should be 0.5 character width between any other element on the same horizontal plane
* The three elements should be horizontally centered at the top of the page

**The following changes to age.html should be made**

* **There are two differences from home.html with respect to style**
  + The text box should **not** expand and contract when it comes into focus
  + The text box should have a width of 5 characters

Consistency in style is relatively easy to achieve and it makes your UI much nicer to use/follow.

Hint:

* You might find Flexbox helpful for arranging your elements

## Task 3

In this task you will create an application for rolling a die. Main page of your app should have a button for rolling the die. When pressed, your JavaScript code should pick a random number between 1 and 6 and request the corresponding image from the server (by redirecting the user to that address). You should use an EJS template engine to send an HTML page with the correct image and a button to roll the die again.

From this task onwards you must always check all user inputs and show an error message if some parameter is invalid (i.e. user has requested an image with a number 7). **You will lose points if you fail to do so.**

Hints:

* res.render() method in Express can be used for rendering template
* res.status() method can be used to set the status code of your response
* You can use express.static() to serve static files such as your JavaScript, CSS and other files

## **IMPORTANT INFORMATION**

Submit all of your sources to the assignment in OMA before the deadline for full points. Assignments submitted even a moment after the deadline will only be able to get half points.

**!!!There is no exception to this rule!!!**