Concordia University
Computer Science and Software Engineering Department
SOEN 287: Web Programming 1 – Winter 2024
Assignment 3

Due Date: By 11:55pm, April 14 (Sunday), 2024
Evaluation: 7% of final mark
Late Submission: none accepted
Type: Individual Assignment
Purpose: The purpose of this assignment is to have you to practice HTML, CSS, JS, Node.js and create a website
CEAB Attributes: This assignments is primarily evaluating your use of Web programming Skills (Knowledge base & design)

Exercise 1

In this assignment, you will become familiar with writing Node.js functions and utilizing pre-existing Node.js functions. All your Node.js functions must be declared in the server file, and each function's name must match the specifications provided below. To demonstrate the functionality of each method, you will call the functions using a form with parameters for the function. By submitting the form, you will call the Node.js server with different paths corresponding to each function. Please note that the use of Global Variables is forbidden for this assignment.

A) Function: findSummation

Parameter(s): A positive integer number (default value is =1)

Given the input positive number (N), find the summation of all positive integer numbers from 1 to N (1+2+3+….+N) and return this summation. If the input parameter is not a number, or if it is not a positive number return false.

B) Function: uppercaseFirstandLast

Parameter(s): String

For each word in the input string, capitalize (uppercase) both the first letter and the last letter and return the modified string.

C) Function: findAverageAndMedian
***Parameter*: An array of numbers.**
Calculate the average and median of the numbers and return both values.

***D)* Function: find4Digits**
Parameter: A string of numbers separated by spaces.
*Return t*he first four-digit number in the string; false if none.

Exercise 2

Create a Node.js page (name it numOfVisits.js) that uses cookies to record and track the number of times a webpage has been visited. If the visitor is looking at the webpage for the first time, it should display the following message: *Welcome to my webpage! It is your first time that you are here.* If the page has been visited more than once, display: *Hello, this is the ## time that you are visiting my webpage* (where ## shows the number of visits). In this case you should also display a message that shows the day of the week, date, time and the time zone on your server that the page was visited last time in the following format (as an example):

Last time you visited my webpage on: Sun Mar 20 17:16:18 EST 2023

(EST= Eastern Standard Time Zone)

Exercise 3:

Write an HTML document to provide a form that collects names and telephone numbers. The phone numbers must be in the format ddd-ddd-dddd. Write a Node.js script that checks the submitted telephone number to be sure that it conforms to the required format and then returns a response indicating whether the number was correct.

Exercise 4: Project

As you know, the last exercise of each assignment is continued from assignment to assignment and will have the goal of building an e-store website by the end of this course.

Adopt A Dog/Cat Website

Running Problem – Version 3

This version requires HTML, CSS, JS and Node.js server. No XML is required.

1) Create a template for your entire site such that you define the header and footer once and include them in each page of the site. This should be done using Node.js.
   The following is just as a reminder of what the header and footer should contain.
   - **Header**: It should include the name of your adoption service and a logo/picture. Clicking on the logo/picture should take you back to the home page. It should also display the current date and time. The Time should automatically refresh every second. (No change from Versions 1 & 2)
   - **Footer**: This should appear at the bottom of all your pages. Give it a distinct background color and include links to a **Privacy/Disclaimer Statement** which can either appear in the content area of the site or in an alert or confirm box. It should display a message promising users that their information will not be sold or misused, and protects the website builder from any incorrect information posted by a pet owner. No change from Version 1).

2) You will manage 3 text files which will reside on the server.
   a. A *login file* which will contain username and passwords. This file will be consulted when a user is logging in to register an animal for adoption or will be written to when a new user registers. Each field should be separated by colon (:) and each user info is in one line. For example if I have a file with 3 accounts, the file will look like so:

      user1:pass1
      user2:pass2
      user3:pass3

   b. An *available pet information file* that will store the information entered by a user when they registered their pet. Details of the information to include in the file and the format are given in the description of **Have a pet to give away** in section 3 below

3) Functionality of your site: Following is a review of each of the entries in your menu with an explanation of the expected behavior some of which

already exists from versions 1 and 2 of this running problem and some which are additions to this last version.

- **Home**: no changes from version 1
- **Browse Available Pets:** is being removed for version 3. We are combining it with the **Find a cat/dog** menu entry.
- **Find a cat/dog:** The form for this option was already created in Version 1, and validated in version 2. When the user presses the Submit button, all records of animals that match the criteria in this form are displayed for the potential pet owner to peruse through.
- **Dog care**: no changes from version 1
- **Cat care**: no changes from version 1
- **Create an account**: This is a new menu entry in Version 3. When a user clicks on this menu option, a login creation page loads into the content area with text fields for a username and one for a password as well as a description of the allowed formats for usernames and passwords. A username can contain letters (both upper and lower case) and digits only. A password must be at least 4 characters long (characters are to be letters and digits only), have at least one letter and at least one digit. Before sending this information to the server make sure that the entered username and password satisfy the format criteria just listed. As usernames are unique, you need to make sure that there isn't already such a user name in use. If the requested username already exists, the server sends back a message requesting a new username/password pair informing the user that this username is not available. If the username is not already in use, then write this new pair to the login file following the format specified in bullet 2 a) and return a message to the content area confirming that the account was successfully created and that they are now ready to login whenever they are ready.
- **Have a pet to give away**: Before a user can create an entry for a new pet on the site they must first log in. Here is the behavior of this option:
    - When a user clicks on this menu option, they will first be asked to login. The client side needs to make sure that the format of both the username and password entered satisfy the criteria listed under the bullet **Create an account** above.
    - When the user clicks the Submit button, the server will check by reading the login file that this login pair exists. If it is a registered login/password pair a new session is started and the form, which was already created in version 1, is loaded into the content area.

If the pair does not exist, a message is send back to the browser saying that the login failed. A user can try as many times as they want to login.

- Once they have successfully logged in, they are ready to fill out the form and submit it.
- Once a form is submitted, the information is added to the *available pet information file*. The 1st entry in the file for each pet must be an integer, which acts as a counter and a unique id for that entry as well as the username of the pet owner, followed by the information from the form. Each entry should be delimited by a colon (:) and each pet record should be in one line. For example if there are 3 pets available for adoption the file would look as follows
  1:user1: cat:persian:0.5:male: …….
  2:user2:dog:5:female: ……
  3:user2:dog:10.5:male:……
  where ….. Represents the rest of the information

- **LogOut**: This option terminates a user's session. The server should return and display a confirmation message.
- **Contact us**: no change from version 1

Hints:

1. Set Up the Project:

- Create a new Node.js project and set up the necessary files and folders.

- Install any required dependencies, such as Express.js, using npm.

2. Create the Server File:

- In the server file (e.g., `server.js`), import the necessary modules and set up the basic server using Express.js.

3. Implement the Node.js Functions:

- Define the following Node.js functions in the server file as specified:

A) Function: findSummation

Parameter(s): A positive integer number (default value is 1)

Description: Given the input positive number (N), find the summation of all positive integer numbers from 1 to N (1+2+3+...+N) and return this summation. If the input parameter is not a number or if it is not a positive number, return false.

B) Function: uppercaseFirstandLast

Parameter(s): String

Description: For each word in the input string, capitalize (uppercase) both the first letter and the last letter, and return the modified string.

C) Function: findAverage_and_Median

Parameter: An array of numbers.

Description: Calculate the average and median of the numbers and return both values.

D) Function: find4Digits

Parameter: A string of numbers separated by spaces.

Description: Return the first four-digit number in the string; return false if none.

4. Handle Requests and Function Calls:

- Set up appropriate routes to handle different paths corresponding to each function.
- When a specific path is requested, call the corresponding Node.js function with the provided parameters.
- Return the results of the function as the response to the client.

5. Create HTML Forms:

- Create separate HTML forms for each function call.

- Each form should contain input fields for the required parameters of the function.

6. Handle Form Submissions:

- Set up event listeners in the HTML forms to handle form submissions.

- Upon form submission, send the form data (parameters) to the appropriate server path corresponding to the function being called.

7. Display Results:

- After submitting each form, handle the server's response and display the results on the client-side.

8. Test the Functionality:

- Test each function by entering different inputs through the forms and verifying the results displayed on the client-side.

Remember to break down the tasks into manageable steps and follow best practices for writing clean and organized code. Test your Node.js functions thoroughly to ensure they work as expected. Feel free to use the “express” library that you find helpful for this assignment.

Note:

1) Although it is not a functional requirement, your site should always be well organized with properly named files and well defined directories.
2) Any pictures, graphics, text, code, you “borrow” from other sites, be sure to give credit to. Be aware of plagiarism.

Submitting Assignment 3

- Please give meaningful names to each html and js file to make the feedback process easier.
- Naming convention for zip file: Create one zip file, containing all folders/files for your assignment.
- The zip file should be called *a#_studentID*, where # is the number of the assignment and *studentID* is your student ID number. For example, for this third assignment, student 123456 would submit a zip file named a4_123456.zip

Evaluation Criteria for Assignment 3 (20 points)

Exercise 1 – 3 pts.	
a) findSummation() b) uppercaseFirstandLast c) findAverage_and_Median d) find4Digits	1 pt 1 pt 1 pt 1 pt
Exercise 2 - 4 pts.	
a) Creating cookie b) Storing the visits count using cookies c) Showing the visits count and date & time	1 pts 2 pts 1 pt
Exercise 3 - 2 pts.	
a) Creating a form b) Data validation on the form	1 pt 1 pt
Exercise 4 - 10 pts.	
a) Find a cat/dog (match criteria requested by user) b) Create an account c) Login procedure d) Session activation when logon e) Have a pet to give away f) Logout procedure	2 pts 2 pts 1 pts 1 pts 2 pts 2 pts
TOTAL	**20 pts**