Assignment 1: Javascript

COMP 315: Cloud Computing for E-Commerce

March 5, 2024

1 Introduction

A common task in cloud computing is data cleaning, which is the process of taking an initial data set that may contain erroneous or incomplete data, and removing or fixing those elements before formatting the data in a suitable manner. In this assignment, you will be tested on your knowledge of JavaScript by implementing a set of functions that perform data cleaning operations on a dataset.

2 Objectives

By the end of this assignment, you will:

- Gain proficiency in using JavaScript for data manipulation.
- Be able to implement various data cleaning procedures, and understand the significance of them.
- Have developed problem-solving skills through practical application.

3 Problem description

For this task, you have been provided with a raw dataset of user information. You must carry out the following series of operations:

- Set up a Javascript class in the manner described in Section 4.
- Convert the data into the appropriate format, as highlighted in Section 5
- Fix erroneous values where possible e.g. age being a typed value instead of a number, age being a real number instead of an integer, etc; as specified in Section 6.
- Produce functions that carry out the queries specified in Section 7.

Data name	Note
Title	This value may be either: Mr, Mrs, Miss, Ms, Dr, or left blank.
First name	Each individual must have one. The first character is capitalised and the rest are lower case, with the exception of the first character after a hyphen.
Middle name	This may be left blank.
Surname	Each individual must have one.
Date of birth	This must be in the format of DD/MM/YYYY.
Age	All data were collected on 26/02/2024, and the age values should reflect this.
Email	The format should be [first name].[surname]@example.com. If two individuals have the same address then an ID is added to differentiate them eg john.smith1, john.smith2, etc

Table 1: The attributes that should be stored for each user

4 Initial setup

Create a Javascript file called *Data_Processing.js*. Create a class within that file called *Data_Processing*. Write a function within that class called *load_CSV* that takes in the filename of a csv file as an input, eg *load_CSV*("*User_Details*"). The resulting data should be saved locally within the class as a global variable called *raw_user_data*. Write a function called *format_data*, which will have no variables are a parameter. The functionality of this method is described in Section 5. Write a function called *clean_data*, which will also have no parameters. The functionality of this method is similarly described in Section 6.

5 Format data

Within the function *format_data*, the data stored within *raw_user_data* should be processed and output to a global variable called *formatted_user_data*. The data are initially provided in the CSV format, with the delimiter being the ',' character. The first column of the data is the title and full name of the user. The second and third columns are the date of birth, and age of the user, respectively. Finally, the fourth column is the email of the user. Ensure that the dataset is converted into the appropriate format, outlined in Table 1. This data should be saved in the JSON format (you may use any built in JavaScript method for this). The key for each of the values should be names shown in the 'Data name' column, however converted to lower case with an underscore instead of a space character eg 'first_name'.

6 Data cleaning

Within the function *clean_data*, the data cleaning tasks should be carried out, loading the data stored in *formatted_user_data*. All of this code may be written within the *clean_data* function, or may be handled by a series of functions that are called within this class. The latter option is generally considered better practice. Examine the data in order to determine which values are in the incorrect format or where values may be missing. If a value is in the incorrect format then it must be converted to be in the correct format. If a value is missing or incorrect, then an attempt should be made to fill in that data given the other values. The cleaned data should be saved into the global variable *cleaned_user_data*.

7 Queries

Often, once the data has been processed, we perform a series of data analysis tasks on the cleaned data. Each of these queries are outlined in Table 2. Write a function with the name given in the 'Function name' column, that carries out the query given in the corresponding 'Query description'. The answer should be returned by the function, and not stored locally or globally.

Function name	Query description
most_common_surname	What is the most common surname name?
average_age	What is the average age of the users, given the values stored in the 'age' column? This should be a real number to 3 significant figures.
youngest_dr	Return all of the information about the youngest individual in the dataset with the title Dr.
most_common_month	What is the most common month for individuals in the data set?
percentage_titles	What percentage of the dataset has each of the titles? Return this in the form of an array, following the order specified in the 'Title' row of Table 1. This should included the blank title, and the percentage should be rounded to the nearest integer using bankers rounding.
percentage_altered	A number of values have been altered between *formatted_user_data* and *cleaned_user_data*. What percentage of values have been altered? This should be a real number to 3 significant figures.

Table 2: The queries that should be carried out on the cleaned data

8 Marking

The marking will be carried out automatically using the CodeGrade marking platform. A series of unit tests will be ran, and the mark will correspond with how many of those unit tests were successfully executed. Your work will be submitted to an automatic plagiarism/collusion detection system, and those exceeding a threshold will be reported to the Academic Integrity Officer for investigation regarding adhesion to the university's policy https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf.

9 Deadline

The deadline is 23:59 GMT Friday the 22nd of March 2024. Late submissions will have the typical 5% penalty applied for each day late, up to 5 days. Submissions after this time will not be marked. https://www.liverpool.ac.uk/aqsd/academic-codes-of-practice/code-of-practice-on-assessment/