IMY 220
Practical 2
NodeJS and NPM

Due: Friday 16 August @ 8:30

The submission instructions are available on ClickUP. Any deviation from these instructions will cause 10% deduction from your mark.

Instructions

- The aim of this practical is to use NodeJS to read and write files.
- Ensure that you include your student details in all worked-on files (as a comment).
- For this practical, you are not allowed to write any loops. You will have to make use of the correct array functions to get the functionality to work. If you do not make use of array functions, you will receive lower marks compared to what you would get if you made use of array functions.

Notes

- Dates are represented in the following format: yyyy/mm/dd (year/month/day).
- You can assume that the JSON in the *events.json* file is valid and correct JSON.
- You will need to initialise a new NodeJS project.

Section #1 - File Reading and Writing

Create a file called *fileManager.js*. This file will contain the functionality for reading and writing files.

- Create a function called *fileRead()*, which takes in the contents of *events.json*. The function should read the data from the file and parse it into JSON.
- Create a function called *fileWrite()*, which takes in new file content. You should write the provided content to a file called *newEvents.json*. The content written to newEvents.json should be in JSON form.
- Export both of these functions for use in other Javascript files.

Section #2 - Data Validation

Create a file called *dataValidation.js*. This file will contain all the functionality for handling the data.

- Create a function called *checkDate()*.
    - This function should take a date (as a string) as its only parameter.

    - It should return true or false depending on whether the date falls between September 9th and September 21st.
    - The first date (September 9th) should be exclusive and the second date (September 21st) should be inclusive.
- Create a function called *checkName()*.
    - This function should take a name (as a string) as its only parameter.
    - The function should return **true** if the name **DOES NOT** contain any special characters.
    - The function should return **false** if the name **DOES** contain any special characters.
    - Special characters include any non-letter or non-number characters (including but not limited to "! @ # $ % ^ & *").
- Export both of these functions for use in other Javascript files.

Section #3 - Displaying Results

Create a file called *index.js*. This file will contain all the main logic for the NodeJS application. Where appropriate, you should only use the existing functions in your fileManager.js and dataValidation.js files.

- Using the *fileRead()* function to fetch the data, use the array functions on the fetched data to filter out all events that do not have a valid date.
    - A valid date is when it returns true when calling the *validDate()* function and passing through the event date.
- Once you have filtered out the invalid events, you will need to go through all the filtered events and check if it has a valid name.
    - A new variable (called *validName*) should be added to the event object and should store the result when you pass the event name to the *checkName()* function.
- Once you have completed the steps above, write the data to a new file using the *fileWrite()* function.

Example Output

If provided with the following *events.json* file:

```
[
    {
        "title": "Picnic",
        "description": "Picnic in the park",
        "date": "2024/09/11"
    },
    {
        "title": "Rock @ concert",
        "description": "Rock concert at the football stadium",
        "date": "2024/09/21"
    },
    {
        "title": "Golf day!",
        "description": "Golf day at the local golf course",
        "date": "2024/09/09"
    }
]
```

The *newEvents.json* file should look like the following:

```
[
    {
        "title": "Picnic",
        "description": "Picnic in the park",
        "date": "2024/09/11",
        "validName": true
    },
    {
        "title": "Rock @ concert",
        "description": "Rock concert at the football stadium",
        "date": "2024/09/21",
        "validName": false
    }
]
```

Submission Files

- index.js
- fileManager.js
- dataValidation.js

Submission Instructions

- ZIP all submission files (using a tool like 7Zip or WinRar) into a ZIP file called *uXXXXXXXX-P2.zip*, where uXXXXXXXX is your student number.
- Submit your ZIP file before 8:30 to avoid any late submissions.