CMPS 350 Web Development Fundamentals - Spring 2022
Lab 11- Asynchronous JavaScript

Objective

The objective of this lab is to practice JavaScript asynchronous programming using **Callbacks**, **Promises** and **async/await**.

Overview

We will collaboratively build the solution of each step after you attempting it by yourself. Then the instructor will demonstrate and explain the model solution. At the end of this exercise, you should be able to use **callbacks**, **promises** and **async/await** to read/write files. The lab has three parts,

- Part A: Practice exercise on callbacks,
- Part B: Practice exercise on promises and
- Part C: Practice exercise on async-await

Part A – Callbacks

Open the file **project1.js** in the attached folder. You will find the following code:

```
//Synchronous code. Change it to async using a callback.
import {fs} from 'fs';
//Synchronous code. Change it to async using a callback.
let data = fs.readFileSync('data/student.json');
console.log(JSON.parse(data));
```

Implement the following tasks in the same order.

a) Convert this code to asynchronous form using a separate callback function.

b) Change the callback function to an anonymous one.

c) Take care of error handling in the callback function.

2) Nested Callbacks

Open the file **project2.js** in the attached folder.

Implement the following tasks in the same order using callbacks.

a) We need to read data from two files: **course.json** and **staff.json**. Both using callbacks.

b) We need finally print all courses with their corresponding instructor names.

   i) Instructor name can be found at the staff file.
   ii) Use **staffNo** in **staff.json** property to match the instructorId from course.json

c) Create two functions **getCourses** and **setInstructorNames**.

```
function getCourses(cb)
```

```
function setInstrctorNames(courses , cb)
```

d) Instructor names are set as a new property to the course object in the **setInstrctorNames** function.

e) Your output is expected to be similar to the following:

```
{
 crn: 107,
 courseCode: 'GENG 107',
 courseName: 'Engineering Skills and Ethics',
 semester: 'Fall 2015',
 instructorId: 12,
 instrctorName: Abdulahi Hassen'
},
```

Part B – Promises

Implement the following tasks in the same order.

a) **Switch to using fs-extra instead of fs.**

- **To install: npm install fs-extra**

b) **In project3.js :** Rewrite the code you created in **Part-1** using **promises**.

c) **In project4.js :** Rewrite the code you created in **Part-2** using **promises**.

3) Chaining Promises

a) **In project5.js :** Reuse the code you created in **project4.js** adding a new promise function to find the number of students in each course.

i. You need to use the student.json file to find the courseIds.

ii. Match it with crn property from the course.json file.

iii. Use map and reduce.

b) Your output should be similar to the following:

```
{
    crn: 300,
    courseCode: 'GENG 300',
    courseName: 'Numerical Methods',
```

```
  semester: 'Fall 2017',
  instructorId: 33,
  instrctorName: 'Saleh Alhazbi',
  studentCount: 50
},
```

Part C –Async/Await

a) **In project6.js :** Rewrite the code you created in **project4.js** using **async/await**.

b) In project7.js : Rewrite the code you created in **project5.js** using **async/await**.