WEB322 Test 2 Name: Junhee Kim

# Instructions: Complete all questions in the spaces provided. This quiz is worth 10% of your final mark and you will have exactly 30 minutes to complete it.

# Question 1 (2 Marks):

Write JavaScript code to convert user input to a nrgative number and assign the number to variable **num** by calling the **toNegativeNumber** function. If the **input** is invalid, console log error message, e.g. " Error: the input is not a number!".

function toNegativeNumber(x) {

if(isNaN(x)) { throw new Error("not a number!") };

if(Number(x) >= 0) { throw new Error("0 or a negative number!") };

return Number(x);

}

let num = NaN, input = document.getElementById("#input").value;

try{

num = toNegativeNumber(input);

}catch(ex){

console.log(“the input is ” + ex.message);

num = Nan;

}

# Question 2 (3 Marks):

Write a JavaScript function named **guessMyAssignmentMark**. In the function, a random mark between 0 and 9 will be generated by using Math.floor(Math.random() \* 9). The function returns a new “Promise” that will “reject” with message “is too bad” if the mark if below 4 and “resolve” with the mark otherwise. Then call the function to console log the “resolved” mark or the “rejected” message.

function guessMyAssignmentMark(){

var randomNum = Math.floor(Math.random() \* 9);

return new Promise(function(resolve, reject){

resolve(randomNum);

if(randomNum < 4){

reject(“is too bad”);

}

}

}

# Question 3 (5 Marks):

Given a server.js file that uses express (i.e. var app = express();), write code to respond to the routes below as specified in the description. **NOTE**: all callback functions **must be written** in the new ES6 **"Arrow Function"** syntax:

1. Assuming our server.js is using the "**path**" module, match the "**GET**" route "**/students/welcome**" and send the file "**source.html**" (located at the root of your application folder). (HINT: do not forget to include **\_\_dirname** in your solution) [1.5 Marks]

app.get(“/students/welcome”, (req, res) =>{

res.sendFile(path.join(\_\_dirname + “/source.html”);

}

1. Match the "**GET**" route "**/studentList**" and return a JSON formatted string: **{message: "all students"}.** Additionally, this route will support optional queries: [1.5 marks]
   * "/**studentList**?**program**=**CPA**" which returns a JSON formatted string: **{message: CPA}**
   * "/**studentList**?**school**=**ICT**" which returns a JSON formatted string: **{message: ICT }**

app.get(“/studentList”, (req, res) => {

res.json({message : res}

}

1. Match the "**GET**" route "**/student/studentID**" and return a JSON formatted string: **{message: studentID}** where **studentID** is the value, **e.g. 041066050,** passed in the **url.** [1 Marks]

app.get(“/student/studentID”, (req, res) => {

res.json({message : studentID})

}

1. Match the "route" that is invoked if **no other paths are matched** (you may assume that it is located below the other route definitions). Ensure that a **404 status code** and the **plain text** message: "**Page does not exist**" are send back. 1 Marks]

app.get(“\*”, (req, res) => {

res.status(404).send({message : ‘Page does not exist’})

}