WEB322 Test 3

# 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 (5 Marks):

1. Recreate the html form (right) using valid HTML according to the following specification: [2 marks]

* Your form MUST submit using "POST" to "/addStudent"
* You are ONLY required to code the form <form>…</form> part, not entire document. No <label> element is required and you may assume that the CSS is created already.

<form method=”POST” action=”/addStudent”>

<fieldset>

<p>Student Number</p>

<input class="form-control" id="studentNum" type="text" />

<p>First &amp; Last Name</p>

<input class="form-control" id="name" type="text" />

<p>Program</p>

<select size="3">

<option value=”1” >CPA</option>

<option value=”2”>CPD</option>

<option value=”3”>CNS/CTY</option>

</select>

<hr/>

</fieldset>

<input type=”submit” value=”Add Student”>

</form>

2. Fill in the gaps in the following server.js code to process data posted from above form, including the code of using middleware to process the form’s POST submission. Once receive the form, the server should return a JSON string in the following format: [3 marks]

{message: "Student: fullName (studentNum, Pogram: program) successfully added"}

Note: fullName, studentNum, and program represent the values entered by user.

var express = require("express");

var app = express();

var multer = require("multer");

var HTTP\_PORT = process.env.PORT || 8080;

app.use(express.static('public'));

const storage = multer.diskStorage({

destination: "./public/addStudent",

filename: function(req, file, cb){

cb(null, Date.now() + path.extname(file.originalname));

}

});

const upload = multer({storage: storage});

app.post("/addStudent", upload.single("student"), (req,res)=>{

const formData = req.body;

const return\_string = "Student: " +

JSON.stringify(formData.fullName)+"("+

JSON.stringify(formData.studentNum) + ", Program: " +

JSON.stringify(formData.program) + ")" + "successfully added";

res.json(return\_string);

})

app.use(express.urlencoded({ extended: true }));

app.listen(HTTP\_PORT, () => { console.log("Express http server listening on: " + HTTP\_PORT); });

# Question 2 (3 Marks):

Assume a Handlebars file receives data of an array that consists of objects with properties:

* employeeId (string)
* fullName (string)
* isManager (boolean)

Write valid HTML code (using Handlebars) to show the ***employeeId***, ***fullName*** values and text “Manager” (only when isManager value is true, otherwise show nothing) for **each object** in the **array** using an **html table**.

The array of objects has been passed to the view using the variable "data". If "data" does not contain any objects, show message "**No data**" in a single table cell.

<table border='1'>

<tr><td>Employee ID</td><td>Full Name</td><td>(is)Manager</td></tr>

<tbody>

{{#each employees}}

<tr>

<td>{{employeeId}}</td>

<td>{{fullName}}</td>

{{#if isManager==true}}

<td>Manager</td>

{{/if}}

</tr>

{{/each}}

{{#unless employees}}

<tr>

<td colspan="3"><strong>No data</strong></td>

</tr>

{{/unless}}

</tbody>

</table>

# Question 3 (2 Marks):

Write a function that describes a custom handlebars block helper that is used to "wrap" it's contents inside a bootstrap row element with one column, i.e:

<div class="row">

<div class="col-md-12">

...

</div>

</div>

This function can have any name you wish - you may assume that it will be added to the handlebars configuration later on.

wrap: function(options){

return '<div class="col-md-12">' + options.fn(this) + '</div>'

}