Assessment of
Web Script Programming
2016-2017

University of Portsmouth

Unit Co-ordinator	Dr Rich Boakes <rjb@port.ac.uk>
Issued	2016-09-29
Code	Web Script Programming / U21264
Purpose	This document describes the work you must complete for the Web Script unit. There are three parts to the assessment.

Schedule & Deliverables

Item	Weighting	You will receive details of your challenge on...	You will submit your work by...	You will receive your marks by...
Client-side Coursework	25%	2016-11-02	2016-11-04	2016-12-02
Server-side Coursework	25%	2016-12-14	2016-12-16	2017-01-23
Web Application	50%	2016-09-28	2017-03-31	2017-05-03

Notes

- This is individual work.
- If you need help you can **ask Rich, Jacek, Matt**. You should also ask your peers first. Discussing problems gives you valuable experience.
- We hope you enjoy these challenges!

Details

Client-Side Coursework

- You will be provided with a part-finished **browser app**.
- The work you must undertake is defined as a series of unit-tests (reflecting the industry practice of test-driven development).
- Your task is to complete various functions and make the unit-tests pass, thus enabling the app to work.
- You shall develop your code in a single JavaScript file which will contain any and all necessary code.
- The coursework is strictly time limited: it will be issued during the lecture in week six, and your code must be submitted within 48 hours of the end of the lecture.
- Submission will be automatic. We will copy all files from a folder called **`webscript1`** on your virtual machine.
- If the folder is not there, or if it is present but empty, or if it is present but diff shows no change from the code we provide, we will consider that a non-submission.

Server-Side Coursework

- You will be provided with an part-finished **server app**.
- The work you need to undertake is again defined as a series of unit-tests.
- Make the unit-tests pass, thus enabling the app to work.
- The coursework is strictly time limited: it will be issued during the lecture in week twelve, and your code must be submitted within 48 hours of the end of the lecture.
- Submission will be automatic. We will copy all files from a folder called **`webscript2`** on your virtual machine.
- If the folder is not there, or if it is present but empty, or if it is present but diff shows no change from the code we provide, we will consider that a non-submission.

Web Application

- You will build an application from scratch, following a brief below.
- The coursework needs to be submitted by the given deadline.
- Submission will be automatic. We will copy all files from a folder called **`webscript3`** on your virtual machine, and we will review the application running on your virtual machine.
- The application must start, and be reachable via HTTP on port 80, by typing **`npm run dashboard`** in the **`webscript3`** folder.
- If the server does not start, we will assess the work based on the source code without the benefit of seeing it run.
- The **`webscript3`** folder should include a **`README.md`** file that explains key features, how to use them, and details your design and implementation rationale.
- If the folder is not there, or if it is present but empty, we will consider that a non-submission.

Web Application Brief

Your challenge is to create a ***Configurable Dashboard***.

You are to specify and construct this configurable dashboard using a combination of HTML, CSS, JavaScript and (optionally) PHP, and any backing store you desire (e.g. a database).

Example use cases include:

- in the home (perhaps for helping a family see what the weather will be like, their upcoming schedule, the latest tweets from their friends, etc.), updated at appropriate intervals.
- in an office (such as the BK first floor, where we have a flat panel for displaying information to staff and visitors). A combination of live data from the web and various databases, stories from various sources, images and video may all need to be displayed.

Configuration of your dashboard may be achieved by editing config files, or through a Dashboard Editor created as part of your deliverable artifact – this is your choice.

Your task is to show (through your work) the extent to which you have met the learning outcomes for the unit.

The learning outcomes (as defined in the Unit Spec.) are:

1. Identify industry best practices in web application design (e.g. client, server and API layers).
2. Design a contemporary web application using industry best practices.
3. Critically evaluate the design and implementation of web applications.

Marking Scheme

The Coursework will be graded using academic judgement, with the following rubric used as a guide.

Topic	Description	Weighting (out of 50)
Functionality	How appropriate is the design? Does it all work? How much does it do? How much is your own work as opposed to libraries?	10
Maintainability	Code style, comprehensibility and maintainability. This includes formatting, file structure, naming - everything that can help your work live on and be useful *after* it is graded, including how well the code and any documentation communicates any concepts necessary to understand the architecture and configuration of the system.	10
Usability	Ease-of-use of your system, including the use of event-driven input, background refresh, drag and drop, intuitive UI design, etc.	5
Accessibility	The appearance of your pages, including use of CSS and relevant capabilities such that the product is suitable for a diverse audience.	5
Reflection	Marked for insight, analysis and evaluation of encountered designs and technologies.	10
Invention	We will award up to 10 bonus marks for unusual qualities, strengths, creativity and invention not otherwise prescribed here.	10