Assessment of Web Programming 2020

Module Coordinator	Dr Rich Boakes <rjb@port.ac.uk>
Issued	January 2020
Code	U30237
Weighting	100%
Submit work by	2020-06-01 23:00 BST on Moodle
Late deadline	2020-06-15 23:00 BST
Feedback by	20 working days after submission

UNIVERSITY OF PORTSMOUTH

Notes and Advice

- The Extenuating Circumstances procedure is there to support you if you have had any circumstances (problems) that have been serious or significant enough to prevent you from attending, completing or submitting an assessment on time.
- ASDAC are available to any students who disclose a disability or require additional support for their academic studies with a good set of resources on the ASDAC moodle site
- The University takes plagiarism seriously. Please ensure you adhere to the plagiarism guidelines.
- Any material included in your coursework should be fully cited and referenced in APA format (sixth edition). Detailed advice on referencing is available from http://referencing.port.ac.uk/
- Any material submitted that does not meet format or submission guidelines, or falls outside of the submission deadline could be subject to a cap on your overall result or disqualification entirely.
- If you need additional assistance, you can ask your personal tutor, learning support ana.baker@port.ac.uk and xia.han@port.ac.uk or your lecturers.

Assignment: Build a Web Application

Your challenge is to create a ***Questionnaire Engine*** using a combination of HTML, CSS, and JavaScript, and any backing store you desire (e.g. a database).

Example use cases include:

- A researcher wants to run a survey but cannot use an existing third party solution (such as SurveyMonkey) due to restrictions on where data is stored. They therefore need a local questionnaire engine to run within their organization.
- A small company wishes to challenge SurveyMonkey by creating a better questionnaire service.

Core Requirements

The system should have the following components (at minimum):

- A client application for participants
  - that makes it possible to fill in a questionnaire
  - that supports a linear flow of questions
  - that works on mobile
- A server
  - that serves the client application
  - that uses a JSON file for structuring the questionnaire (the format and an example file will be supplied in class and through Moodle)
  - that stores the responses (submitted by the client through an API)
  - that supports a download of the responses in a structured way for the questionnaire author to use (e.g. JSON or CSV).

Optional Features

Possible additional features include:

- A client page that allows questionnaire authors to create/edit questionnaires through a GUI rather than by writing JSON by hand.
- A client page that aggregates the results and shows them to the questionnaire author. This could be through graphs, averages etc.
- Support for more complex types of question (eg. Likert scales, matrices, sliders), with a specification for the necessary extensions to the supplied JSON questionnaire file format.
- Support for branching, so optional questions can be shown based on the answers of previous questions.
- Security (authorization) based on OAuth authentication.

Learning Outcome Focus

Remember that 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 module 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.

Delivery requirements

1. The application must start, and be reachable via HTTP on port 8080 when run on your VM (or installed on ours, which is the same spec).
2. **Libraries may only be used on the server.** They must be specified in `package.json` for installation via `npm install`
3. If your application requires a configuration step (e.g. to setup a database), this must be achieved via the command `npm run setup` – we will not undertake any other manual steps.
4. The application must launch with `npm start`
5. If the server does not start, we will assess the work based on the source code without the benefit of seeing it run.
6. You must include a `README.md` file that explains key features, how to use them, and details your design and implementation rationale.

Marking Scheme

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

Topic	Description	Weighting
Functionality	How appropriate is the design (data, code, architecture)? Does it all work? How much does it do? How much is your own work as opposed to libraries?	25
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.	25
Usability	Ease-of-use of your system, including the use of event-driven input, background refresh, drag and drop, intuitive UI design, etc.	15
Accessibility	The appearance of your app, including use of CSS and relevant capabilities such that the product is suitable for a diverse audience.	5
Delivery	Does it install and start as required?	10
Invention	We will award up to 20% bonus marks for unusual qualities, strengths, creativity and invention not otherwise prescribed here.	20