Interactive Web Applications

cct | College Dublin
Computing • IT • Business

Module Title:	Interactive Web Applications
Assignment Type:	Continuous Assessment 1
Project Start Date:	26/10/2019
Assignment Compiler:	Mikhail Timofeev (cct-iwa-a.slack.com/cct-iwa-b.slack.com)
Weighting:	25%
Due Date:	**17:00, 08/12/2019**
Method of Submission:	GitHub for the code, Moodle for the report
Late submission:	Late submissions will be accepted up to **5 days** after the deadline. All late submissions are subject to a penalty of **10% per day**. Submissions received more than **5 days** after the deadline above **will not be accepted**.
Last document update:	01/12/2019

Module Learning Outcomes Assessed

MLO2 Develop the needs of a web application and implement an intuitive and user-centric interface design that can be applied to web applications to develop customizable user interfaces.

MLO3 Implement the requirements of an interactive web application solution taking into account the framework, user interface and storage needs.

MLO6 Design embeddable services that can integrate simple web applications into hosted web services.

MLO7 Assess the security needs of a web application to aid the development and integration of security measures.

Assignment Description

Choose an area of your interest to design, create and deploy an interactive web app that incorporates:

[30%] Structured data stored in **XML** and/or **JSON** formats, where these files can be validated (**XSD/JSON Schema**) and transformed to **HTML** for client-side consumption.

[30%] Retrieval and deletion of this data on a server-side with **Node.js** and through the use of appropriate **modules.**

[20%] Clean, intuitive and responsive user interface (**Bootstrap** or similar) and appropriate visual representation of the data on the client-side (**CSS**, **JavaScript/JS frameworks**).

[10%] Security measures, such as validation of user input on the client-side (**JavaScript/JS frameworks**) and sanitisation of user input on the client-side (**Node.js**) through the use of appropriate **modules**.

[10%] The finished application should be accompanied by a **report** (5 pages max) that outlines the idea behind your application, describes each of the technologies used and includes a short **user manual** (10%). The cover sheet for the report should include a **full student name**, **student number**, **project title** and the project's **GitHub repository link**.

Additional Information

- This project is an **individual project** and you have all the freedom to work on a topic of your interest.
- The application should be developed using an online IDE, such as **Gitpod.io** (the free tier includes 100 hours a month for developing/debugging/running you app), and all of your code from day one should be stored in your personal **GitHub** repository that is traceable, i.e. shows an **adequate number of commits** that go back to the very start of your project.
- All of the code and the report will be checked for **plagiarism**, so make sure **everything is referenced** through the use of comments. **The code that is not referenced properly will not be marked and will be reported for plagiarism**.
- **Outside of special circumstances, any code submitted to GitHub after the deadline – will not be marked**.
- Any other **technical details** that might come up during the course of this project should be clarified directly with your lecturer.
- A **grading rubric** for each of the grade items is outlined below. Please use this as a guide for the development of this project.

	Data **(30 marks)**	**Server-Side** **(30 marks)**	**Client-Side** **(20 marks)**	**Security** **(10 marks)**	**Report** **(10 Marks)**
<40	Data fails validation and is poorly structured with insufficient levels of complexity, no HTML conversion	Data fails to pass from server to client	Minimal scripting, lack of UI	No security implemented on either the client or server sides	Badly formatted and referenced report addressing only a few of the elements required, demonstrating no analyses and consideration of technologies
40 − 49	Data validation returns some errors and the data is somewhat structured with minimal complexity, some form of HTML conversion is present	Data just passes from server to client-side with no CRUD functionality	Minimal client-side scripting with limited UI presence	At attempt of implementing security measures is visible on either the client or server sides	Poorly formatted and referenced report addressing only a few of the elements required, demonstrating little analyses and consideration of technologies
50 − 59	Data validates with no errors, is adequately structured with minimal complexity, HTML conversion is present	Data passes from server to client-side with limited CRUD functionality	Adequate presence of client[side scripting with an appropriate UI	Limited security is present on either the client or server sides	Well formatted and referenced report addressing most elements required, demonstrating adequate analyses and consideration of technologies
60 − 69	Data validates with no errors, is well structured with several levels of complexity, HTML conversion is present	Data passes from server to client-side, most of the CRUD functionality is present	Well optimised client-side scripting with a robust UI	An appropriate level of security measures is in place for both the client or server sides	Well formatted and referenced report addressing all elements required, demonstrating good analyses and consideration of technologies
70 +	Data validates with no errors, is excellently structured with many levels of complexity, HTML conversion is present	Data just passes from server to client-side and full CRUD functionality is present	Fully optimised client-side scripting that provides rich visual effects for the UI, UI is well-designed	Well-designed and rigid implementation of security measures on both the client or server sides	Excellently formatted and referenced report addressing all elements required, demonstrating excellent analyses and consideration of technologies