SENG365 2025 Web Computing Architectures: Assignment 2 Implement Web Client

1 Summary of submission requirements

Course coordinator:	Ben Adams
Set date:	First week of semester.
Submission date:	**11:59pm, Friday, 23 May 2025**
Late submission:	**NO late submission is allowed** for this assignment (without Special Consideration or similar)
Items to submit:	Submit zipped project to Learn.
% of overall grade:	30% of the course grade

2 Objectives

The objective of the second assignment is to develop a web client that communicates via the already-specified API with the web server developed from Assignment 1. The web client should satisfy the list of user stories provided via Learn. Taken together, the web client and the web service, with the API, provide a petition site.

3 Required technologies

For this assignment, you must use React to the version levels that are used in the labs. We will provide a standardized web service (on a VM test server) so you should not develop the web client based on your own web service from Assignment 1. (We will explain how to access the web service – e.g. the IP address or URI – in due course.)

You may choose to use additional JavaScript libraries etc., depending on your implementation, provided these libraries are sourced from a recognized repository (content delivery network), and provided these packages do not break dependencies on React, or on the Node.js and Express versions in the labs.

You do not have to develop your applications only in the labs, you may use your own computing resources. You should maintain your source code under Git version control.

4 Indicative criteria used in the assessment

In this assignment your commitment to the user experience should be front-and-centre. This doesn't mean flashy UI features or "look-at-me" interfaces, though.

The requirements are provided in the form of User Stories in the "Assignment 2 back stories” file. Implement as many as you can (each story counts in the assessment).

As stated in the introduction, your app must run in a prescribed way from a lab workstation (see below) to be marked, and we will assess using the installed version of **Chrome**.

We shall be assessing:

1. Correctness and ease of use for each implemented story when run against an instance of the reference Assignment 2 API implementation (a buggy or broken story won't improve your grade). **This is the main component of assessment.**

2. Robustness and stability. An app that crashes or fails will be hard to assess. We'll also expect that your application will perform simple field validations and access checks.

3. Visual appeal and general creativity ('going the extra mile').

5 Compulsory lab final week for testing

We intend to use a version of crowdsourced testing to contribute to the assessment of Assignment 2. Each submission will be 'exercised' by a randomized set of students during the final week of labs.

6 Submission

Submit your assignment, as one zip file, to the Learn Assignment 2 'dropbox'. Before you submit, delete your `node_module` directory (to save a lot of space) and then zip up your project as one file. You must name your zip file according to the following format `<usercode>.zip` (e.g., abc123.zip).

Your app must be able to be run using the following procedure:

1. Unzip the submission to the `/local`[1] directory on the lab workstation.
2. cd directory
3. `npm install`
4. `npm run start`
5. Open browser to appropriate port

The api-v1-implementation API that your app is to use will be provided closer to the time. If your submission cannot be run when we assess it, your grade for the assignment will be affected.

[1] This is important because it returns latency when installing node packages from an external repository.