Advanced Internet Programming Assessment 3

Motivation

The best way to learn is by doing. The best way to learn **how to build** a web application is to **build** a web application.

Today, technologies are evolving so rapidly that an important part of building a web platform lies in the selection of appropriate frameworks and independently learning how to use that technology.

We will be supporting you throughout the whole process of developing a web application: helping you make decisions, guiding you through your independent learning and teaching you general principles of good design. These general principles will apply not only today but also decades later when you are developing or developing with new technologies that haven't been invented yet.

This assessment is motivated by learning objectives 1, 2, 3, 4, 5, 6 and 7 from the subject outline.

Assessment Item

You will work in groups of up to three developers to design and develop a modern web application (groups of three are strongly recommended). You will use JavaScript (running on the server and in the browser) to develop a web application of your own choosing. Time during the Week 2 tutorial will be allocated to help you form groups and discuss assignment plans.

You will demonstrate the full **functionality** of your application in the Week 11 laboratory session, but you will have a chance to make changes to your code for final submission at the end of Week 12.

Your application may do anything you wish (except for to-do list or address book). I encourage you to have fun. Design a web application that you think would bring people happiness or joy, or build something that would be genuinely useful to you or somebody you care about.

Your application is subject to the following requirements:

- All code must be written in JavaScript to run in NodeJS and in the browser (extensions of JavaScript such as JSX, TypeScript and Flow are also fine)

- It must have a mechanism for users to sign up and log in (passwords must be stored securely, not in plain text)
- Your server-side code must make use of a database to save and retrieve data (the database should not be directly exposed to the end-user)
- It must expose a RESTful API
- It must be deployed on the public internet using a cloud host (e.g., Amazon, Google Cloud, Azure)
- You must use git to track your changes and you must regularly push your commits to a **public repository** so that your tutor and classmates can see your work
- Your application must **not** use jQuery
- Your application must **not** be a to-do list or address book
- Your application must **not** be offensive (no hate, insults, discrimination, pornography or violence)

Here are some ideas to get you started, but you should choose something that you care about:

- Online multiplayer rock-paper-scissors game
- Dating site for developers
- Pokémon tracker
- Wedding gift registry
- Family event calendar
- Social network for IT students
- Study group finder
- Job advertisement board
- Restaurant booking system
- Social news site (e.g., a simple 'Hacker News' clone)

You will **not** be assessed on the quality of the idea. The idea **does not have to be 'good'** so long as you implement it well (i.e., using patterns and principles covered in this subject).

Don't over-complicate your system: it should be sophisticated enough to demonstrate the assessment requirements but you should focus on demonstrating good design, as opposed to demonstrating that you can do lots of work.

Your system will be evaluated in three ways:

- **Demonstration of functionality:** Does it work? Is it reliable, secure, fast, scalable and user friendly?
- **Design:** Have you made good design choices? Are you layering your code? Are you using external modules appropriately? Would a professional developer be happy to contribute to your project without feeling like they need to rewrite large portions of it?

◿ **Code and Documentation:** Is your code well documented? Are your git commits useful? Is your code well written? Do you have useful file, method and variable names? Do you have a useful README?

Costs

BitBucket, Github and Gitlab all provide free hosting for public projects.

Please use trial periods and free non-commercial licenses when integrating with commercial services or deploying your code. Reimbursement or financial support will NOT be provided if you choose to use paid plans or services that cost money.

Please be very careful when deploying your application. Many cloud hosting providers provide free levels of service in addition to education plans suitable for deploying your application. These free services should be used for this assignment. Please be careful when deploying instances as running additional instances can quickly result in a large bill.

Submission

Your assignment will be demonstrated to your tutor and (some) of your classmates during your Week 11 lab session. You will demonstrate the functionality and get some feedback on your source code. You will have a chance to polish your source code before final submission at the end of Week 12 (Friday 19 October 11:59pm).

Before you demonstrate your system, you should thoroughly test your system:

◿ Attempt to log in with an incorrect password
◿ Enter invalid data and check that validation works on all fields
◿ Test all the functionality you have implemented: check every button and link works
◿ Check that you can log out and log in as somebody else
◿ Attempt to access secure pages without logging in

Assessment Criteria

This assessment is worth 40% of your final grade.

You will be receiving feedback and guidance throughout the semester. Preparation activities and in-class activities for Assessment 1 will be related to this Assessment.

The final submission of your project will be assessed according to the following criteria:

Correctness/completeness of your implementation (10 marks)

This criteria will be marked in-class during your Week 11 tutorial.

0	Your application does not run, crashes regularly or fails to satisfy several required functions or non-functional requirements.
2	You have implemented most but not all of the required features, you have not satisfied all of the non-functional requirements or your application crashes during demonstrations (or shows user-unfriendly error messages).
6	You have implemented all of the features and satisfied the non-functional requirements. Your application does not crash even with unexpected inputs.
10	Your application is reliable, secure, fast, scalable and user friendly. Your application is ready for public use.

The quality of your design (20 marks)

This criterion will be used to mark your final submission.

0	Your system lacks structure or coherent design.
5	You code has an appropriate design but there is substantial room for improvement. In particular, there may be significant mixing of purposes (e.g., user interface state embedded in data storage code).
10	You code is appropriately designed. You have separated your system into layers and you have used libraries where appropriate. Some refactoring would improve the quality of what you have done.
15	Your system is well designed and code is well separated into layers. Components are reusable. The codebase represents the quality expected from professional developers. You have used libraries where appropriate.
20	Your system is beautifully designed and flawless and demonstrates elements of novelty and innovation. You have elegantly used libraries where appropriate.

The quality of your documentation (10 marks)

This criterion will be used to mark your final submission.

0	Your code is difficult to read or it would be difficult for another developer to join your team. You will also receive this mark if your code is messy, there are large amounts of unused/commented-out code or if you have not used revision control.
5	Your code and the documentation is acceptable. Most code has comments. Classes, methods and variables have clear names. A skilled professional developer would be able to contribute to your project without feeling a need to rewrite major parts of what you have done.
7	The quality of your documentation and the clarity of your code (including commits, READMEs and the code itself) is of a high standard. Your documentation makes it easy for other developers to contribute to your code. There is no unnecessary duplication of code.
10	The quality of your documentation and the clarity of your code (including commits, READMEs and the code itself) is of an extremely high standard. It makes it easy for professional developers of varying skill levels to understand the evolution of the system as well as the current state of the system. Technologies are used appropriately and idiomatically (i.e., using standard coding conventions for the technology).

Assessment Penalties

The system demonstration and final submission are marked separately. A 10% penalty will apply to the relevant criteria for each day your submission is late. Late submissions should be uploaded directly to UTS Online.

If you are not present for the system demonstrations in Week 11 and you should contact the subject coordinator to arrange an alternate demonstration. Your demonstration is due at the start of the tutorial and so is considered late if it is not ready to be demonstrated at the start of your laboratory session.

Late submission penalties are calculated based on multiples of 24 hours from the due date.

Group Work and Misconduct

This is a group assignment. You are encouraged to discuss your work with other groups. However, all other work that you submit as part of this assignment must be entirely your own and any assistance properly identified and acknowledged. You are permitted to make use of libraries and other resources found online but you **must clearly identify the**

source at the top of every single method that you did not write. Your git commit logs should be an accurate reflection of the code that you have contributed (i.e., if a team member is not able to use git, teach them how to use git rather than committing their changes).

It is expected that group members will contribute equally to the assignment. The assignment will be marked as a group, and then individual marks are allocated on the basis of a peer assessment process.

Please discuss with your tutor if a group member is not contributing to the assignment (or if you are not given opportunities to contribute). Your tutor may, on the basis of contributed workload, heavily penalize students who do not contribute to the assignment. A student who contributes nothing may receive zero marks for the assignment. The use of revision control is mandatory and will assist your tutor in understanding the contribution of each group member.

Based on consultation with group members and/or an analysis of the revision control logs, your tutor or subject coordinator may override peer assessment to more fairly reflect contributions.

If you are experiencing group problems, do not wait until the "last minute". Discuss with your tutor as soon as possible so that there is sufficient time to correct any issues or negotiate a solution.

If you are in doubt, please ask your tutor. Please refer to the faculty's handbook for more information about Student Misconduct:
http://www.uts.edu.au/sites/default/files/FEIT%20Student%20Guide.pdf