NWEN304 Group Programming Project

This project is worth 40% of your total grade (25% individual + 15% group)

1. Important dates

Group member preference entry: Sunday 16 September 2018 (https://goo.gl/forms/W8EO0Z9qvDJxk3ds1)

- Please use the online form to let us know whom you do NOT want to work with in the group project.
- If you are happy to work with any students in the class, you do not need to provide any input in the online form.
- We will try to form groups according to your preference but may not be able to satisfy all preferences from all students.
- We cannot take into account your preference of working particularly with any students in the group project.

Group formation: Tuesday 18 September 2018 (or Wednesday 19 September)

Project Progress presentation: Friday 12 October 2018 (3pm to 5pm)

Group presentation and interview: Last two days of the examination period (to be confirmed)

Project submission: Last day of the examination period (to be confirmed)

2. Project Overview

In this project, you will develop a RESTful web application by using NodeJS. For example, you can choose to develop an online coffee shop, an online bookstore, online shoe store, etc. You can be creative and develop an application that sounds more interesting to you. It might be helpful if you could talk to the course lecturers with regard to the web application that you want to develop and get some feedback.

To get some idea about the application to be developed, we recommend you to have a look of (https://www.oberlo.com/blog/ecommerce-website-design-examples) for some examples. These example web applications support several key functionalities that are essential to your project, such as user registration, user login, view and purchase specific items, etc. You can consider using Boostrap (https://getbootstrap.com/) to develop the frontend of your web application. However, the aesthetic aspect of your application frontend is not essential to this project since our aim is to develop secure, scalable and reliable web server applications. Therefore, using Bootstrap is completely optional.

You can also download a tutorial project from our project page (https://ecs.victoria.ac.nz/Courses/NWEN304_2018T2/Projects). In the tutorial project, you can find some HTML, CSS, JavaScript and image resources for building the frontend of a simple online shopping application. You are allowed (but not mandatory) to use these resources directly for your own project. However, you are NOT allowed to use PHP scripts, database resources and related artifacts provided in the tutorial project to build the server side of your application.

This group project has a strong focus on developing a secure, scalable and reliable RESTful web server application. Therefore, you need to build your RESTful server completely by yourself (including setting up the databases for your server application).

Groups will be allocated for you (you can expect to work in a group of 3/4 students). You can provide a list of up to 3 students whom you do not wish to work with. We will try to avoid grouping them with you. For this purpose, please express your preferences in here (https://goo.gl/forms/W8EO0Z9qvDJxk3ds1).

All code must be version controlled using Git and follow the rules specified in the *General Git Requirements* and *General Programming Requirements* (see below). Information from the repository may be used in grading.

3. General Git Requirements (mandatory)

- All source code must be version controlled in a single main repository for each group.
- Every group member has write access to the repository.
- Commits done in pairs must feature the term "peer programming". Other commits must not (this is to make them easily searchable).
- Repository can be stored on a *public Github* or *GitLab* hosted by our school. Please grant us access to your repository as soon as it is created.
- Work must be committed to the repository on the day that it is done.

4. General Programming Requirements (mandatory)

- Your web application should work properly in the ECS unix lab machine versions of Chrome and Firefox.
- Test scripts should be provided (e.g. a list of curl commands) to verify the correctness of the server side of your Web application.
- Use cases should be provided to verify the correctness of the browser side of your web application.
- You are only allowed to use open source libraries in your project. You should also demonstrate good understanding of any open source libraries used (e.g. you can justify clearly the validity of using any open source libraries).
- You must build your server application by using NodeJS and host your application on Heroku.

5. Important Requirements for Your Project

In this project, there are several important requirements that your application should support. These requirements are highlighted below and will be carefully assessed for grading.

Basic requirement (65%)**:**

C1. Clearly define and correctly implement a RESTful web application.

a. Implement basic user registration functionality. Your application must be able to control the complexity of the password chosen by the user (ideally although not compulsory, some visual feedback on password complexity can be provided to the user while the user is entering his/her new password).
b. Implement basic user login / logout functionality.
c. Support easy search and browsing of shopping items (or any form of resources

specific to your application).

C2. Host your server successfully on a cloud platform, i.e. Heroku, so that your RESTful web application can be easily accessed from anywhere in the Internet.

C3. Correctly set up a database (including carefully designed table structures) at the server side to achieve data persistency in your application.

Completion requirement (15%):

B1. Use OAuth / OpenID Connect and online service (e.g. Google Identity Platform or Facebook Login) for user registration and authentication.

B2. Proper implementation of salted password hashing in the web server for secure user account management. Please refer to our project page (https://ecs.victoria.ac.nz/Courses/NWEN304_2018T2/Projects) for more guidance.

B3. Implement a simple user password reset function through a web-based interface.

B4. Your web application must be able to track users' activities (e.g. to provide a shopping cart functionality in your online shopping application and to remember users preferences).

a. Registered users can review their purchase/order history in the application.
b. Administrators can review and modify users' purchase/order history.
c. Administrators can archive the purchase/order history for any user. That means the history will be saved in a local file at the server side and the user will no longer be able to review the past purchase/order history.

B5. Demonstrate proper use of caching related HTTP headers in your server application.

a. Understand the use of some important caching related HTTP headers (to be covered in lectures).
b. Implement these HTTP headers properly in your server application.
c. Provide several test cases for verifying the correct use of caching related HTTP headers.

Challenge requirements (20%):

A1. Implement a simple recommendation service to recommend relevant resources (e.g. new products in an online shop) based on information provided by external services (e.g. Yahoo Weather) and/or user purchase/order history and/or user account information. This will require keeping extra information in the database, collecting geolocation information from the user and using an external service. Your recommendation algorithm can be very simple. The focus here is to enable your web application to use user information in conjunction with external services.

A2. Your web application should be designed and implemented with clear consideration of the *privacy aspect*.

A3. Support a timeout function: when users stop using the application for a certain period of time, they need to login again.

A4. Implement email loop for password reset. Particularly, if the user requires resetting his/her password, your web application should send an email that contains a random single-use token to the user. The token is strongly tied to the user account and is

embedded in a password reset link included in the email message. When the user clicks the password reset link containing a valid token, your application will allow the user to specify a new password. The token can only be used a single time.

A5. Evaluate and measure the performance of your server application.
   a. Measure the response time of each API function provided by your server application under varied workload.
   b. Show the variation of response time under varied workload by using Bar charts (to be included in the final group presentation/demonstration).
   c. Evaluate the performance of the database at the server side. Determine whether the database stands for a performance bottleneck of your web application (to be included in the final group presentation/demonstration).

6. Marking

The project will be marked out of 100%. The following describes the distribution of marks.

- Individual
  - Project progress presentation (5 minutes): 5%
  - Project interview: 55%
- Group
  - Group project presentation and demonstration (20 minutes): 40%

Progress presentation will be 5 minutes and will be conducted in Week 11. Group project presentation, demonstration and individual interview will be arranged in the examination period.

7. Things for submission

You should submit the following through the online submission site.

1. A readme file explaining:
   a. How to use your system
   b. What the REST interface is
   c. What error handling has been implemented in your system

2. A zip of your entire project directory, including the source for web pages and RESTful server application.

3. The test cases for frontend and the test scripts (e.g. a list of curl commands) for the server end of your web application.

4. The test cases for verifying the correct use of caching related HTTP headers and the corresponding test results.

5. A short document that summarizes your database design and briefly explains your database access/management code.