DALHOUSIE
UNIVERSITY

FACULTY OF
COMPUTER SCIENCE

# CSCI 4177/5709 — Advanced Topics in Web Development

# Group Project

# 1. Overview

The final project for this course is to be completed in groups of 4-6 students, **assigned by the instructor**. Once groups are assigned, **no changes in student-group allocation will be made**. This project involves the creation of a web application through the use of Client-Side and Server-Side languages, web development Frameworks, and other application and development concepts and techniques discussed in this course. Groups will be expected to assess the suitability of specific techniques, approaches, and tools for the successful completion of this project.

For this project, **you will be required to submit** the following group deliverables: **Project Proposal, Project Report**, and a **Web Application** which groups will **Demo**. Finally. As part of your project, you will also be expected to individually submit a **Peer Evaluation** in which you will determine how many points (out of 15) to award to your group mates based on their performance in your group. In the following sub-sections, you will find the list of requirements for each deliverable of your project.

***Purpose.*** The purpose of these deliverables are to test your comprehension of the various concepts discussed in class, and your ability to apply them to solve a given problem.

***Grades.*** Each deliverable will be graded out of 100 points, and will be scaled to 10, and 20 points for the project report, and demo of the prototyped application, respectively.

***Software / Code Editors.*** Coded deliverables must be completed **without** the aid of "visual" website generating software. This includes desktop programs such as Dreamweaver or web based programs such as Wix. You can use tools such as Notepad++ / Vi / Vim / Sublime Text, Visual Studio Code, etc.

***Submission.*** All deliverables must be submitted on Brightspace (**https://dal.brightspace.com**) and Git Lab (**https://git.cs.dal.ca**).

***Late Submission Policy.*** Late deliverables are **not** accepted. However, no penalty will be assessed for deliverables that are late due to documented situations (See Syllabus).

***Academic Integrity.*** Dalhousie academic integrity policy applies to all submissions in this course. You are expected to submit your own work. Please refer to and understand the academic integrity policy, available at: **http://www.dal.ca/dept/university_secretariat/academic-integrity.html**

***Content for the website.*** Do not copy and paste content from any websites into your prototype application. You will have to create your own content to include on your website.

## 2. Project Report

**[15% Group Deliverable]**

For the third deliverable for this project, you will be expected to submit a Project Report and a Prototype Application. Section 2 will focus on the Learning Objectives, Requirements, Grading Rubric and Submission procedures for your written report. The requirements for your Prototype Web Application are specified in Section 3.

For your Project report, you will be expected to use the Project Report Template available on the course's Brightspace site, this template is an adaptation from the IEEE/PES Technical Report template[1]. In this report your group will have an opportunity to document the ins and outs of your high-fidelity prototyped application, as well as explain and justify any modifications made to the overall application since your group's Project Proposal submission. **Your report's grade is determined by both a written (50%) and a programming (50%) component.**

### Learning Objectives:

1. Apply technical writing skills to describe the development techniques, approaches and skills that were implemented into the development High-Fidelity prototype of a web application.
2. Defend design and development approaches implemented, as well as reflect on changes made to a previously specified proposal.
3. Build a functional a web application by implementing and customizing suitable libraries, plug-ins, frameworks ad APIs for a project, give its specifications, features, purpose, and overall expected functionality.
4. Collaborate with others in completing a usable and functional prototype of an e-commerce web application by applying the development skills learned in this course.

### Requirements:

**D3.1.** Use the **Project Report Template (.doc)** provided to you through **Brightspace** and follow the instructions detailed in the template.

***Note:*** *You will need to submit your Project Report in* ***PDF*** *file format. Your Report must include your group project's URL. There will be a* ***15% deduction*** *for any submission not matching the required file format. There will also be a* ***5% deduction*** *for any submission not matching the required format specification (i.e., not following the template).*

[1] https://www.ieee-pes.org/images/files/doc/tech-council/PES-Technical-Report-Template_Jan_2016.docx

**D3.2.** Provide a completed **Project Log** form.

*__Note:__ The __Project Log__ form is included in your Project Report Template file.*

**D3.3.** Your paper must have a list of references of the items/sources used in the preparation of the report, using the **ACM or IEEE citation styles**.

*__Note:__ You may use the __Computer Science Research Guide__ (__http://dal.ca.libguides.com/c.php?g=257109&p=1717772__) for guidance in citation styles used in Computer Science. You may change your citation style from the one chosen for your Project Proposal. However, it is strongly recommended for you to keep your citation style consistent throughout a specific document.*

**D3.4.** Your Project Report should be divided into the following sections:

- **Title Page:** This includes the title the paper, student identification, and email address.
- **Abstract:** A brief 100-word summary of what the paper is about.
- **Keywords:** Provide up to 10 keywords (in alphabetical order) to help identify the major topics of the paper.
- **Introduction:** Should set the stage for discussion and motivate the topic. The introduction should set the scope of your project.
- **Background:** Should describe any jargon, and knowledge that the reader is expected to have. i.e., A description of your project (e.g., service or product), explaining your project's background, purpose, objectives and/or goals.
- **Application Details:** A description of your application, including the following information for your application:
    - **Target User Insight:** A short description of your target user base (i.e., user personas,), assumptions on why users would use this particular application (i.e., scenarios, use cases, user flow or task flow diagrams), a description of any requirements or pre-requisites that users must fulfil or have in order to be able to use your application (i.e., specific knowledge, device, required training).
    - **User-Centered Design Approach:** Explain how your user insights were taken into consideration or used in the design and development approach for your application (i.e., Information Architecture, sitemapping, wireframming, design and layout).

*__Note:__ Ensure you provide justifications for your design decisions and implementations. As previously mentioned, you are welcomed to use any material you may find useful from a previous assignment, but you are encouraged to address any issues mentioned in that assignment's feedback.*

- **Application Workflow:** Describe the overall application workflow for your project in regards to your interaction design approach to describe the front-end of your application, as well as the back-end processes and/or services in your application:

  - **Interaction Design:** A description of how your application's front-end is meant to work. How are processes triggered and handled?
    Provide graphs or figures that illustrate how the front-end of your application's *processes and services* work (i.e., click streams, user task flow diagrams), more importantly, show try to illustrate the support needed from the back-end.
    Provide a completed *use case* for each of the tasks in the feature you have chosen for this assignment, your use cases must include both normal and alternate flows. Use cases must also define the scenario relevant to the specific use cases as well as identify any user personas for whom the application is intended.

  - **Process and Service Workflow:** a description of the overall workflow of your application, before focusing on the details of how the back-end of your application (in regards to the ONE feature you have chosen) is meant to work and support the front-end. How are processes, for your chosen feature, triggered and handled? Provide graphs or figures that illustrate how the backend of your application processes and services work (e.g., workflow diagrams). A diagram detailing the expected file and folder structure for your intended features.

***Note:*** *In this section, you will be expected to illustrate your application's overall workflow, as well as your chosen feature's workflow from a front-end and back-end perspective. Ensure you provide justifications for your design decisions. You are essentially expected to explain how the back-end of your application (e.g., Process Workflows) is meant to or expected to support the front-end of your application (e.g., Task Flow Diagrams) as defined in your project proposal.*

- **Conclusion:** It is particularly important that you state clearly what you have done, not merely what you plan to do. Please consider what the reader will learn from your paper, and how they will find your work useful.

- **Recommendations:** Any recommendations you would make for continuous support to this application or further development (i.e., what would you recommend a fellow developer do or consider when adding more features to this application?).

- **References:** A list of references of the items/sources used in the preparation of the report.

- **Appendices:** Additional sources (if needed) that provide further supportive information to your report but may not be necessarily required.

**D3.5.** Provide the URL through which the website can be accessed.

***Note:*** *The Instructor and Markers will not grade any individual assignment or deliverable not integrated with the group's master. As well, all coded work must be live and accessible to the markers, remotely through a web browser.* ***Local deployments for marking purposes will not be accepted.***

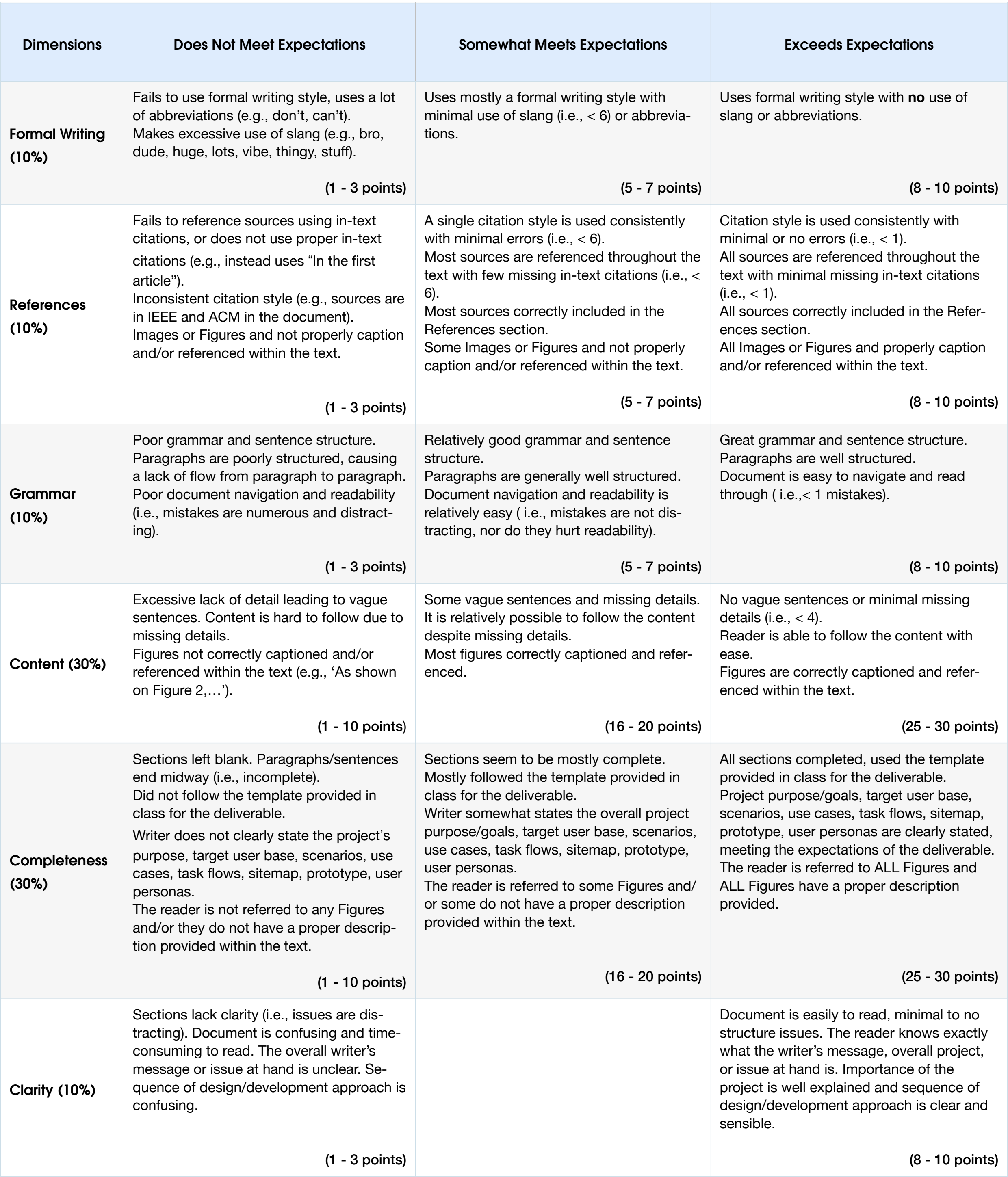
**D3.6.** The submission file for the written report must be named **D3_Group#.pdf and submitted electronically via Brightspace ONLY.**

**D3.7.** The written portion of your Project Report will be graded using the rubric shown on **Table 1** (see Marking Rubric Section)**.**

## Marking Rubric

The marking rubric shown in **Table 1** will be used for marking your Project Report:

**TABLE 1. CSCI 4177/5709 RUBRIC FOR WRITTEN WORK**



| Dimensions | Does Not Meet Expectations | Somewhat Meets Expectations | Exceeds Expectations |
|---|---|---|---|
| **Formal Writing (10%)** | Fails to use formal writing style, uses a lot of abbreviations (e.g., don't, can't).<br>Makes excessive use of slang (e.g., bro, dude, huge, lots, vibe, thingy, stuff).<br>**(1 - 3 points)** | Uses mostly a formal writing style with minimal use of slang (i.e., < 6) or abbreviations.<br>**(5 - 7 points)** | Uses formal writing style with **no** use of slang or abbreviations.<br>**(8 - 10 points)** |
| **References (10%)** | Fails to reference sources using in-text citations, or does not use proper in-text citations (e.g., instead uses "In the first article").<br>Inconsistent citation style (e.g., sources are in IEEE and ACM in the document).<br>Images or Figures and not properly caption and/or referenced within the text.<br>**(1 - 3 points)** | A single citation style is used consistently with minimal errors (i.e., < 6).<br>Most sources are referenced throughout the text with few missing in-text citations (i.e., < 6).<br>Most sources correctly included in the References section.<br>Some Images or Figures and not properly caption and/or referenced within the text.<br>**(5 - 7 points)** | Citation style is used consistently with minimal or no errors (i.e., < 1).<br>All sources are referenced throughout the text with minimal missing in-text citations (i.e., < 1).<br>All sources correctly included in the References section.<br>All Images or Figures and properly caption and/or referenced within the text.<br>**(8 - 10 points)** |
| **Grammar (10%)** | Poor grammar and sentence structure.<br>Paragraphs are poorly structured, causing a lack of flow from paragraph to paragraph.<br>Poor document navigation and readability (i.e., mistakes are numerous and distracting).<br>**(1 - 3 points)** | Relatively good grammar and sentence structure.<br>Paragraphs are generally well structured.<br>Document navigation and readability is relatively easy ( i.e., mistakes are not distracting, nor do they hurt readability).<br>**(5 - 7 points)** | Great grammar and sentence structure.<br>Paragraphs are well structured.<br>Document is easy to navigate and read through ( i.e.,< 1 mistakes).<br>**(8 - 10 points)** |
| **Content (30%)** | Excessive lack of detail leading to vague sentences. Content is hard to follow due to missing details.<br>Figures not correctly captioned and/or referenced within the text (e.g., 'As shown on Figure 2,…').<br>**(1 - 10 points)** | Some vague sentences and missing details. It is relatively possible to follow the content despite missing details.<br>Most figures correctly captioned and referenced.<br>**(16 - 20 points)** | No vague sentences or minimal missing details (i.e., < 4).<br>Reader is able to follow the content with ease.<br>Figures are correctly captioned and referenced within the text.<br>**(25 - 30 points)** |
| **Completeness (30%)** | Sections left blank. Paragraphs/sentences end midway (i.e., incomplete).<br>Did not follow the template provided in class for the deliverable.<br>Writer does not clearly state the project's purpose, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas.<br>The reader is not referred to any Figures and/or they do not have a proper description provided within the text.<br>**(1 - 10 points)** | Sections seem to be mostly complete.<br>Mostly followed the template provided in class for the deliverable.<br>Writer somewhat states the overall project purpose/goals, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas.<br>The reader is referred to some Figures and/or some do not have a proper description provided within the text.<br>**(16 - 20 points)** | All sections completed, used the template provided in class for the deliverable.<br>Project purpose/goals, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas are clearly stated, meeting the expectations of the deliverable.<br>The reader is referred to ALL Figures and ALL Figures have a proper description provided.<br>**(25 - 30 points)** |
| **Clarity (10%)** | Sections lack clarity (i.e., issues are distracting). Document is confusing and time-consuming to read. The overall writer's message or issue at hand is unclear. Sequence of design/development approach is confusing.<br>**(1 - 3 points)** | | Document is easily to read, minimal to no structure issues. The reader knows exactly what the writer's message, overall project, or issue at hand is. Importance of the project is well explained and sequence of design/development approach is clear and sensible.<br>**(8 - 10 points)** |

## Submission Guidelines

Your research paper must be submitted through **Brightspace ONLY**.

**To submit your work to Brightspace:**

- Convert your Project Report into a PDF prior to submission.
- Your submission file for your Project Report must be named **D3_Group#.pdf**

  **Note: You will need to submit your Project Report in PDF file format. There will be a 10% deduction for any submission not matching the required file format. There will also be a 5% deduction for any submission not matching the required format specification (i.e., not following the template) or file naming convention.**

- Submit your Project Report (in PDF) on **Brightspace**.

# 3. Report Prototype Application Requirements

**[Group Deliverable]**

In order to complete this project you will be required to complete a prototype of a web application, as defined by your group in your **Project Report**. To aid you in brainstorming for your Project Proposal, the following sections provide you with the expected technical requirements of your fully functioning prototype.

**Learning Objectives:**

1. Break complex tasks into smaller more manageable parts and steps, in order to delegate these tasks within the group based on roles and responsibilities.
2. Work in groups to complete complex problems, through new and/or collaborative development approaches
3. Learn to test various components of an e-commerce web application prototype.

**Application's Requirements:**

For your prototype application to be considered complete, you must meet the following requirements:

The requirements for your Project Proposal's **programming deliverable are as follows:**

**D3.8.** Implement **UX Principles** and **UI design principles,** discussed in class. These principles must be relevant to the overall use of your application.

**D3.9.** Your application MUST be responsive, i.e., it should be supported by more than one device. If your application is specifically meant to be used on a single type of device (e.g., only mobile, only desktop), you are expected to define the layouts and/or widths that will be supported and will still need to implement a **responsive UI**.

***Note:*** *Whether your application will support one or more than one device, ensure that you include a* ***justification in your proposal and report****. Why implement a responsive UI if your application will only support one type of device (e.g., laptop and/or desktop)? Because we now live in a world where it is possible to have a laptop or desktop device that supports touch interaction, therefore* ***your application is expected to be useful, usable, and appropriately respond to the device being used by a given user.***

**D3.10.** Apply relevant **development optimization techniques** to your application's web assets, such as its HTML, CSS, JS, PHP, third party scripts, libraries, development frameworks, APIs, plug-ins, and/or other media, in order to increase your application's efficiency.

***Note:*** *You will be expected to customize any third party scripts, development frameworks, APIs, and/or any plug-ins implemented in your web application. Further, you will also be expected to include the proper documentation in your README file. Using these components as is, without any valid justification and/or documentation for doing so will result in a maximum possible grade (for the programming component of this deliverable) of 60%.*

**D3.11.** Apply relevant **Search Engine Optimization (SEO) techniques** to your application's Front-End, and implement the necessary files for optimized web crawling (e.g., copy used, sitemap.xml, robots.txt).

**D3.12.** Implement APIs that you have assessed to be beneficial and/or relevant to your application's functionality. You **will be expected to justify** your decision for using a specific third party scripts, development frameworks, libraries and APIs in your Project Proposal and Report.

**D3.13.** If applying a Front-End CSS Framework (e.g., Foundation, Skeleton, Bootstrap), you MUST **ensure you have significantly customized any frameworks** in order to implement it into your project (i.e., you cannot simply implement a Framework 'as-is').

***Note:*** *Frameworks MUST be* ***significantly customized*** *in order for your group to receive credit for this portion of your project. Simply changing colours or fonts will be considered a 'significant' change or customization effort. Using styling frameworks as is will automatically result in a 60% grade deduction. Design templates are not allowed to be used for any programming deliverables in this course, doing so will result in an automatic grade of 0 for the programming component of this deliverable.*

**D3.14.** Apply relevant **optimization techniques to your HTTP requests** (e.g., GET, POST, PUT, DELETE) in order to increase the speed of your application and optimize its communication methods.

***Note:*** *Your report MUST include details of how your development approach implemented optimization techniques for the purposes of improving the overall performance of your application.*

**D3.15.** Apply **Object-Oriented Programming techniques** (in the language and/or framework of your choosing), in order to better establish database connections, handle errors (e.g., try/catch statements), work with SQL/JSON or XML data, and increase security measures (e.g., stored procedures, web attack prevention).

***Note:*** *Your code will be marked for quality so excessive commenting, non-meaningful variable/ function names, buggy applications of code, will have a negative impact on your overall project mark.*

**D3.16.** Apply **development techniques** for improving the performance and UI of your application, e.g., asynchronous calls when necessary.

***Note:*** *The use of AJAX techniques and/or asynchronous functionality will also be marked based on your application of usability principles through your code.*

**D3.17.** Implement a useful and **relevant third party API and/or development framework or library** (e.g., Angular.JS, Node.JS, React, CakePHP, PHalcon), suitable for the purpose, functionality or goal of your project, into your application.

**D3.18. Your application is expected to be a secured application (i.e., front-end and back-end) and built with scalability in mind**, i.e., MUST have front-end and back-end security techniques and approaches applied to mitigate most common attacks, e.g., XSS attacks, Session Fixation attacks, Cross-Site Request Forgery, Session Hijacking, SQL injections, Password Sniffing, Brute Force attacks, and other Session Management attacks.

***Note:*** *Your report must include details in regards to the security of your application.*

**D3.19.** Your application **MUST** have a proper **landing page**, informing the user of the purpose and/or benefits of your application, and serve as a **lead generation page**.

***Note:*** *The lack of a Landing Page will result in a 15% grade deduction.*

**D3.20.** Ensure your application is **cross-browser and cross-platform compatible** (i.e., your application can be viewed in a variety of browsers and platforms), as well as **site consistency**.

***Note:*** *Any lack of cross-browser and/or cross-platform compatibility will result in a maximum grade of 50% to your overall project.*

**D3.21.** Your code must follow as much as possible **W3C HTML (http://validator.w3.org) and CSS Validation (http://jigsaw.w3.org/css-validator/) guidelines and principles**.

***Note:*** *Failure to validate will result in a maximum grade of 50%. However, any warnings or errors due to browser proprietary tags and/or selectors will be ignored, as long as no cross-browser compatibility problems are visible.*

**D3.22.** Your project will be expected to make use of some sort of data storage solution, e.g., MongoDB, SQL, suitable for the purpose, functionality and overall goal of your application.

**D3.23.** Your project will be **marked for code quality and appropriate technology**. Distracting animations and graphics, slow loading pages, non-functioning elements, excessive comments, non-meaningful variables and/or function names, redundant logic, and other similar flaws will affect your mark accordingly. While simpler is often better in the context of the WWW, you should not avoid the use of more complex features where they are useful, necessary, or effective. You should strive to create an interesting and attractive website.

***Note:*** *Keep in mind that deductions will be made in cases where the marker encounters broken links or missing expected functionality or items (e.g., front-end validation on a form, landing page, missing expected back-end functionality).*

**D3.24.** Overall, your project will be expected to have a given number of features based on the number of students allocated to your group. i.e., *Group Members x 2 = Total Intended Features* of which you will be expected to complete 70% by your Project Report's due date (i.e., *Total Expected Features).*

***Note:*** *The number of features your project is expected to have is equal to 'Group Members x 2' (i.e., if your group is made up of 5 members, you are expected to have 10 intended features stated on your proposal), from this number, you are expected to complete 70% of these features (i.e., your scope or "Must Have" features in this example would be 7 features).*

**D3.25.** The code created by each of the members of your group will be expected to be fully integrated by this deliverable's due date, live and remotely accessible to the markers through a web browser.

***Note:*** *Your application will be graded using* ***the Project URL you provide in this deliverable (i.e., specified in your Report and README file)****. The Instructor and Markers will not grade any individual assignment not integrated with the group's master.* ***Local deployments for marking purposes will not be accepted.***

**D3.26.** The programming portion of your Project Report will be graded using the rubric shown on **Table 2** (see Marking Rubric Section)**.**

## Marking Rubric

The marking rubric shown in **Table 2** will be used for marking your project's web application:

**TABLE 2. CSCI 4177/5709 RUBRIC FOR PROGRAMMING WORK**

| Dimensions | Does Not Meet Expectations | Somewhat Meets Expectations | Meets Expectations | Exceeds Expectations |
|---|---|---|---|---|
| **Front-End Frameworks (20%)** | Fails to implement front-end frameworks and/or does not provide justification for not using a framework.<br>**(0 - 5 points)** | Implements front-end frameworks but fails to customize its implementation and/or user interaction not well supported.<br>**(10 - 13 points)** | Implements somewhat customized front-end frameworks but frameworks do not properly support user interaction.<br>**(14 - 16 points)** | Successfully implements and customizes front-end frameworks in a way that supports user interaction.<br>**(18- 20 points)** |
| **Back-End Frameworks (30%)** | Fails to implement back-end frameworks and/or does not provide justification for not using a framework.<br>**(0 - 5 points)** | Implements back-end frameworks but fails to customize its implementation and/or front-end not well supported.<br>**(10 - 15 points)** | Implements somewhat customized front-end frameworks but frameworks do not properly support user interaction.<br>**(17 - 22 points)** | Successfully implements and customizes back-end frameworks in a way that properly supports the front-end of the application.<br>**(28 - 30 points)** |
| **Design (10%)** | Overall design is cluttered, without a clear colour palette or typographic style defined. Design elements are not consistent and/or does not reflect requirements set out in earlier deliverables. Fails to implement proper content hierarchy, and/or uses Lorem Ipsum content, and/or design is not responsive.<br>**(0 - 2 points)** | Overall design is somewhat cluttered, has a somewhat clear colour palette but no clear typographic style defined. Design elements are somewhat consistent. Design elements and Content Hierarchy are somewhat consistent. However, Content organization is still somewhat lacking.<br>**(3 - 4 points)** | | Overall design is clean, aesthetically pleasing. Deliverable has a clear colour palette and typographic style is well defined. Design elements are consistent and enhance the usability of the application. Design elements are consistent and enhance the usability of the application.<br>**(8 - 10 points)** |
| **UX & Usability (20%)** | Fails to implement front-end validation and/or user feedback techniques. Deliverable lacks error recovery messages, when needed. Poor consideration for UX and usability in its design and functionality.<br>**(0 - 3 points)** | Implements some front-end validation and/or user feedback techniques, but messages used are inconsistent and not clear on the issue or how to correct it. Somewhat considers UX and usability into its design and functionality.<br>**(7 - 10 points)** | Implements some usable front-end validation and/or user feedback techniques, and though messages provide an idea of the issue and its solution, they are still inconsistent. Better consideration for UX and usability into its design and functionality.<br>**(14 - 16 points)** | Deliverable properly applies usable front-end validation and/or user feedback techniques throughout, with clear and consistent error recovery messages. UX and usability techniques are at the core of its design and functionality.<br>**(18 - 20 points)** |
| **Security (20%)** | The application does not implement ANY front-end and/or back-end security techniques and approaches to mitigate common attacks.<br>**(0 points)** | The application implements some front-end OR back-end security techniques and approaches to mitigate some common attacks.<br>**(4 - 6 points)** | The application implements front-end AND back-end security techniques and approaches BUT ONLY mitigates SOME common web attacks.<br>**(4 - 6 points)** | The application successfully implements front-end AND back-end security techniques and approaches to mitigate MOST common web attacks.<br>**(18 - 20 points)** |
| **README.txt** | Fails to include a README file, or file is empty.<br>**(-20 points)** | README file fails to include complete and/or correct details on code sources and citations, and/or justifications missing URL.<br>**(-10 points)** | | Includes a README file with complete and correct content such as application's URL, code referencing and justifications.<br>**(0 points)** |
| **Code Quality** | Code is not optimized or excessively redundant, causing performance issues. Excessive and/or redundant comments, variables/function names are not meaningful.<br>**(- 25 points)** | | | Code is optimized, no redundancy or any performance issues. Code is clear with succinct comments only when necessary. Variables/function names are meaningful. The logic of the code is clearly defined.<br>**(0 points)** |
| **W3C Compliant** | Fails to be cross-browser and/or cross-platform compatible and/or apply W3C guidelines where appropriate.<br>**(-25 points)** | | | Deliverable is cross-browser and cross-platform compatible, and applies W3C guidelines where appropriate.<br>**(0 points)** |

## Submission Guidelines

Your project report's programming deliverable must be submitted through **Brightspace and Git Lab**..

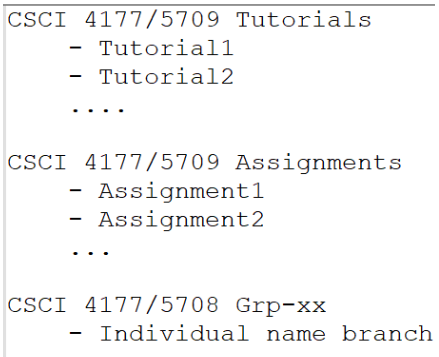
### To submit your work to Brightspace:

- **For your proposal's application submission**, include your proposal's README.txt or README.md. Your README file must provide the appropriate details for the technologies used and work done in developing your proposed application prototype, as well as the prototype's URL

- Your proposal's README file must match naming conventions specified in the Course Syllabus, i.e., it must be named **D3_Group#.txt** or **D3_Group#.md**, **and be submitted into the Proposal Web Application assignment dropbox on Brightspace.**

### To submit your work to Dal's Git Lab:

- First, your group must have a project repository created on Dal's GitLab, with the corresponding individual branches, i.e., each individual group member will have their own individual branch, similar to the one shown on Figure 1. Ensure all your project files are included in your project folder.



```
CSCI 4177/5709 Tutorials
    - Tutorial1
    - Tutorial2
    ....

CSCI 4177/5709 Assignments
    - Assignment1
    - Assignment2
    ...

CSCI 4177/5708 Grp-xx
    - Individual name branch
```

***Figure 1.*** *GitLab Folder Structure Example.*

***Note:*** *For this deliverable, your group will be submitting a single GitLab repository link, along with a single application URL.*

- Setup your project folder as a private project and add the course **Teaching Assistants (TAs) and Instructor** as **'Maintainers'** to your project, using their **CS IDs.**

  ***Note:*** *The CS ID for this course are provided in our Tutorial 2 handout and module. Failure to add the course CS ID as 'Maintainer' for your work on Git Lab will result in a maximum possible grade* *of 50%. Make sure that your README file includes the deployment link for your group's A3 submission and your group's project repo, **failure to do so will result in a grade of 0.***

- While you are free to use GitHub for deployment purposes, do keep in mind that your code repository on GitLab **MUST** be used for grading purposes.

## Deploying your Work:

For the purposes of this assignment, you may use **Netlify and Render, and/or any other deployment option of your choice. Your deployment solution must be agreed upon by your entire project group**. To allow for this flexibility, **your README.txt or README.md file must include the URL from which your assignment can be accessed.**

However, should you choose to use Timberlea, below are a series of instructions to help you out.

- Login to **Timberlea** at **timberlea.cs.dal.ca** **using your CS Username and CS Password**. You may use Terminal or an FTP Client (e.g., FileZilla) to connect to Timberlea.

  ***Note:*** If you are using an FTP Client, you may use **sftp://timberlea.cs.dal.ca** as your hostname. If you need help logging on to Timberlea, please follow the instructions available on the CS Support website (**https://web.cs.dal.ca/~tlin/cs_support/**)

- Once logged into **Timberlea,** go into your **'public_html'** folder and, if you have not already done so, create a folder called **'csci4177-5709'**.

  ***Note:*** All your work **must** be reside inside your '**csci4177-5709**' folder, this folder **must be nested inside your** '**public_html' folder**. If your files are not inside your '**public_html'** directory on **timberlea.cs.dal.ca, the markers will not be able to access your work and you will receive a grade of 0. It is the responsibility of the student to ensure their assignments are available for grading before the due date.**

- Go into your **'csci4177-5709'** folder and create an assignment folder called **'project'**.

  ***Note:*** You will need to create an assignment folder for each individual assignment, as well as your final project, as we go through the term (i.e., a2, a3, A3, and project).

- Place the all the files you created for this assignment inside the **'project'** folder you created on Timberlea.

  ***Note:*** In order for your assignment files to be accessible through a browser for testing and grading, you must ensure you are using the correct file permission settings on your files and folders. On a shared server, such as Timberlea, it is recommended to **use '755' (i.e., rwxr-xr-x) on folders**, and **'644' (i.e., rw-r--r--) on individual files**. You can set your file permissions easily through an FTP client by right clicking on the file or folder you want to set specific permission settings. Depending on your FTP client, you will need to click on **'Get Info'** or **'File Permissions'**. Once on the file permissions window, you can simply enter the numeric value described above.

- Test your assignment is readily accessible and properly working. Your URL will likely include a port address. Ensure you include this URL in your **README.txt** file.

***Note:*** You are encouraged to check your work through the URL specified in your README.txt file, as **the Instructor and TA will not be checking any other URL**. The rule of thumb is "if you can see your assignment on a browser through your assignment's URL, the TA and Instructor can see and grade your assignment". It is the student's responsibility to ensure their submission is accessible and working as expected.

- Using Development Frameworks:

  - If as part of your assignment you plan to use a development framework such as Node or Angular, do keep in ming that you will have to use a custom port when launching your web application. Ports 1000 through 40000 are allowed through the firewall for this purpose.

  ***Note:*** Most students should be able to use their CS ID. However, if you do encounter issues with your account, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

  - If as part of your assignment you plan to use CodeIgnitor, a PHP development framework, you may simply download these files into your public_html directory and serve them from your Timberlea account. CodeIgnitor also includes a database configuration file, so you may need to have your own copy of this file.

  ***Note:*** Should you have any issues, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

  - If as part of your assignment you plan for use .NET, you will have to use a custom port when launching your web application.

  ***Note:*** Should you have any issues, please stop by the FCS Help Desk located on the main level of the Goldberg Computer Science building.

- In addition to the submission instructions detailed above, there are a few other guidelines you should follow for this assignment:

- You ***must*** use HTML5 semantic document divisions (discussed in class) where possible, instead of simply using divisions <div>.

- You ***must not*** copy / paste code from any websites – this amounts to plagiarism. Do not copy / paste text and content from the websites either.

***Note:*** *In the case you find a piece of code that would be useful for a programming assignment, you may be able to use it if you meet the following requirements.*

*Your tutorial/assignment/project must include a* ***README.txt*** *file that specifies the following:*

- *The* ***function and line(s) of code*** *(as noted in a Source Code Editor) that include any content taken from a web source.*
- *The* ***web source*** *(i.e., URL) where the code was taken from and the date on which it was accessed.*
- *A brief* ***explanation*** *of what the code is meant to do in its original form (i.e., as it is shown on the web source),*
- *An* ***explanation of how*** *the original* ***code was modified*** *in order to be used in your tutorial/ assignment/project.* ***You must have extensively customized the code in order to be able to use it, copy/paste or simply re-naming variables will not suffice.***

- ***Images.*** If you want to use other images on your website, be sure to use images that are published under Creative Commons licenses, i.e. you can use them with proper attribution. A good place to search for such images is on the Creative Commons website: **http://search.creativecommons.org/** *Always remember to attribute credit to the image creator. Credit should either be in HTML comments or in a separate document named* ***“README.txt”***

- The emphasis in your Project is for you to apply your knowledge of front-end and Back-end web development techniques for creating usable applications where the front-end successfully supports user interaction.

## 4. Academic Integrity[2]

At Dalhousie University, we respect the values of academic integrity: honesty, trust, fairness, responsibility and respect. As a student, adherence to the values of academic integrity and related policies is a requirement of being part of the academic community at Dalhousie University.

What does academic integrity mean?

Academic integrity means being honest in the fulfillment of your academic responsibilities thus establishing mutual trust. Fairness is essential to the interactions of the academic community and is achieved through respect for the opinions and ideas of others. Violations of intellectual honesty are offensive to the entire academic community, not just to the individual faculty member and students in whose class an offence occurs. (See Intellectual Honesty section of University Calendar)

How can you achieve academic integrity?

- Make sure you understand Dalhousie's policies on academic integrity.
- Give appropriate credit to the sources used in your assignment such as written or oral work, computer codes/programs, artistic or architectural works, scientific projects, performances, web page designs, graphical representations, diagrams, videos, and images. Use RefWorks to keep track of your research and edit and format bibliographies in the citation style required by the instructor (See http://www.library.dal.ca/How/RefWorks).
- Do not download the work of another from the Internet and submit it as your own.
- Do not submit work that has been completed through collaboration or previously submitted for another assignment without permission from your instructor.
- Do not write an examination or test for someone else.
- Do not falsify data or lab results.

These examples should be considered only as a guide and not an exhaustive list.

What will happen if an allegation of an academic offence is made against you?

I am required to report a suspected offence. The full process is outlined in the Discipline flow chart, which can be found at: http://academicintegrity.dal.ca/Files/AcademicDisciplineProcess.pdf and includes the following:

1. Each Faculty has an Academic Integrity Officer (AIO) who receives allegations from instructors.
2. The AIO decides whether to proceed with the allegation and you will be notified of the process.
3. If the case proceeds, you will receive an INC (incomplete) grade until the matter is resolved.

[2] Based on the sample statement provided at http://academicintegrity.dal.ca.

4. If you are found guilty of an academic offence, a penalty will be assigned ranging from a warning to a suspension or expulsion from the University and can include a notation on your transcript, failure of the assignment or failure of the course. All penalties are academic in nature.

Where can you turn for help?

- If you are ever unsure about ANYTHING, contact myself.
- The Academic Integrity website (http://academicintegrity.dal.ca) has links to policies, definitions, online tutorials, tips on citing and paraphrasing.
- The Writing Center provides assistance with proofreading, writing styles, citations.
- Dalhousie Libraries have workshops, online tutorials, citation guides, Assignment Calculator, RefWorks, etc.
- The Dalhousie Student Advocacy Service assists students with academic appeals and student discipline procedures.
- The Senate Office provides links to a list of Academic Integrity Officers, discipline flow chart, and Senate Discipline Committee.