DALHOUSIE UNIVERSITY
FACULTY OF COMPUTER SCIENCE

CSCI 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, 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** a **Project Proposal, Project Report**, and a **Web Application** which groups will **Demo**. Finally, as part of your project, you will also be expected to 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 assignments 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 assignments are **not** accepted. However, no penalty will be assessed for assignments 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

Due: April 4th,
[15% Group Deliverable]

For the second deliverable for this project, you will be expected to submit a Project Report, 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.

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. 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:

For your Project Report you must:

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. 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).

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 proposal, 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.

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

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., students, professionals, developers, age range, location), assumptions on why users would use this particular application, 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, design and layout).
- **Application Workflow:** Describe the 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 front-end is meant to work. How are processes triggered and handled? Provide graphs or figures that illustrate how the backend of your application processes and services work (i.e., click streams, user task flow diagrams, use cases, scenarios, personas).
  - **Process and Service Workflow:** a description of how your back-end is meant to work. How are processes triggered and handled? Provide graphs or figures that illustrate how the backend of your application processes and services work (e.g., workflow diagrams).
- **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 proposal but may not be necessarily required.

D3.5. Provide the URL through which the website can be accessed (if available).

D3.6. The submission file for your proposal must be named **D2_Group#.pdf and submitted electronically via Brightspace ONLY.**

Marking Rubric

As shown in **Table 1**, the following grading criteria will be used for marking your Project Report:

TABLE 1. CSCI 5709 RUBRIC FOR WRITTEN WORK

Dimensions	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
Formal Writing (10%)	Fails to use formal writing style, uses a lot of abbreviations (e.g., don't, can't). Makes excessive use of slang (e.g., bro, dude, huge, lots, vibe). **(1 - 4 points)**	Uses mostly a formal writing style with minimal use of slang (i.e., < 6) or abbreviations. **(5 - 7 points)**	Uses formal writing style with no use of slang or abbreviations. **(8 - 10 points)**
References (20%)	Fails to reference sources using in-text citations. Does not use proper in-text citations (e.g., instead uses "In the first article"). Inconsistent citation style (e.g., sources in IEEE and ACM in the document). **(1 - 9 points)**	Citation style is used consistently with minimal errors (i.e., < 4). Most sources are referenced throughout the text with few missing in-text citations (i.e., < 4). Most sources correctly included in the References section. **(10 - 15 points)**	Citation style is used consistently with minimal or no errors (i.e., < 1). All sources are referenced throughout the text with minimal missing in-text citations (i.e., < 1). All sources correctly included in the References section. **(16 - 20 points)**
Grammar (10%)	Poor grammar and sentence structure. Paragraphs are poorly structured, causing a lack of flow from paragraph to paragraph. Poor document navigation and readability (i.e., mistakes are numerous and distracting). **(1 - 4 points)**	Relatively good grammar and sentence structure. Paragraphs are generally well structured. Document navigation and readability is relatively easy (i.e., mistakes are not distracting, nor do they hurt readability). **(5 - 7 points)**	Great grammar and sentence structure. Paragraphs are well structured. Document is easy to navigate and read through (i.e.,< 3 mistakes). **(8- 10 points)**
Content (40%)	Excessive lack of detail leading to vague sentences. Content is hard to follow due to missing details. Figures not correctly captioned and referenced within the text (e.g., 'As shown on Figure 2,…'). Writer does not clearly state the project's purpose. Writer does not clearly describe their prototype/application design/development process **(1 - 15 points)**	Some vague sentences and missing details. It is relatively possible to follow the content despite missing details. Most figures correctly captioned and referenced. Writer states the overall project purpose. The prototype/design/development approach is justified and described in some detail. **(16 - 30 points)**	No vague sentences or minimal missing details (i.e., < 4). Reader is able to follow the content with ease. Figures are correctly captioned and referenced within the text. Project purpose is clearly stated. The prototype/design/development approach is justified and described in great detail. **(31 - 40 points)**
Completeness (10%)	Sections left blank. Paragraphs/sentences end midway (i.e., incomplete). Did not follow the template provided in class for the deliverable. Scope of project was vague. **(1 - 4 points)**	Few sections left black. Sections seem to be mostly complete. Mostly followed the template provided in class for the deliverable. Scope of project somewhat defined **(5 - 7 points)**	No sections left blank, including those that do not apply to the task or project at hand. All sections completed, scope of the project well defined. Used the template provided in class for the deliverable. **(8 - 10 points)**
Clarity (10%)	Sections lack clarity (i.e., issues are distracting). Document is confusing and time-consuming to read. The overall writer's message is unclear. Not clear what the overall project is about. Unclear what the issue at hand is, or the importance of the project. Sequence of design/development approach is confusing. **(1 - 4 points)**	Document is readable despite occasional structure issues. The content provided in most sections is mostly clear. The reader has an idea of what the writer's message is. Somewhat clear what the overall project is about. Somewhat clear what the issue at hand is, and the importance of the project. Sequence of design/development approach is somewhat clear. **(5 - 7 points)**	Document is easily readable, minimal to no structure issues. The content provided in sections is clear. The reader knows exactly what the writer's message is. It is clear what the overall project is about. It is clear what the issue at hand is, and the importance of the project. Sequence of design/development approach is clear and sensible. **(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 Proposal 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. Prototype Application Requirements

Due: April 4th
[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**.

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 Technical Requirements:

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

D4.1. Implement **UX Principles** discussed in class into the **UI design** of your application, these principles must be relevant to the overall use of your application.

D4.2. Unless your application is specifically meant to be used on a single type of device (e.g., only mobile, only desktop), implement a **responsive UI**.

Note: Should your application only be available on a single device, ensure that you include a **justification in your proposal and report**.

D4.3. Apply relevant **development optimization techniques** to your application's web assets, such as its HTML, CSS, JS, PHP, third party scripts, and/or other media, in order to increase your application's efficiency.

D4.4. 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).

D4.5. Implement HTML5 APIs, that you have judged to be beneficial and/or relevant to your application's functionality, in order to boost its performance (e.g., offline web apps, geolocation, web worker, offline storage).

D4.6. If applying a Front-End 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.

D4.7. Apply relevant **optimization techniques to your HTTP requests** in order to increase the speed of your application and 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.

D4.8. Apply **Object-Oriented Programming techniques** in PHP (or the language 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.

D4.9. Apply **AJAX techniques** for improving the performance and UI of your application.

Note: The use of AJAX techniques will also be marked based on your application of usability principles through your code.

D4.10. Implement a useful and **relevant third party API** (e.g., Angular.JS, Node.JS, CakePHP, PHalcon) into your application.

D4.11. 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.

D4.12. Your code must pass validation by the **W3C HTML (http://validator.w3.org) and CSS Validation (http://jigsaw.w3.org/css-validator/) services**.

Note: Failure to validate will result in a maximum grade of 50%. Validation is important! 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.

D4.13. Your application should be considered to be a **secured application**, i.e., you must strive to make use of client-side and server-side techniques to authenticate users, validate form fields, as well as prevent common web attacks (i.e., XSS attacks, session fixation attacks, cross-site request forgery, session hijacking, SQL injections, password sniffing, brute force attacks).

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

D4.14. 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.

D4.15. 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 Features Expected.*

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 features).

Submission Guidelines

Your assignment must be submitted through **Brightspace and Git Lab**.

To submit your work to Git Lab:

- Create a project folder called **Group#_GroupName**. Ensure all your assignment files are included in your project folder.
- 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 IDs for your course TAs and Instructor will be provided to you during our course tutorial.

To submit your work to Timberlea:

For the purposes of this assignment, you may use **Timberlea**, **Heroku** or **Azure** as your deployment option. **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.**

- 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'**.

  Note: All your work **must** be reside inside your '**csci4177**' 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'** folder and create an assignment folder called **'group#'**.

  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, a4, and project).

- Place the all the files you created for this assignment inside the **'group#'** 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** or **README.md** 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 .

- 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 Assignment 2 is for you to apply your knowledge of identifying elements from the wireframe. i.e., the process of creating a website, from the conceptualized design in a wireframe.

- This approach helps you translate your ideas / concepts from class into actual websites, allowing you to apply what you have learned.

- You are welcome to include additional features in your project. **However, bear in mind the following:**

  - Your submission **must** meet the criteria specified in the Project's Technical Requirements, first and foremost. Beyond this requirement, you are welcome to include additional aspects of future assignments. However, **no bonus points will be granted or replacement will be allowed** for any missing aspects of your project.

  - You stand to lose points if the additional markup / CSS elements that you might implement interferes with the basic requirements of your project.

- I will not stop you from exploring beyond what is taught in class or what is expected in these assignments. However, please be mindful of what you submit as your assignment submission.

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.