DALHOUSIE UNIVERSITY
FACULTY OF COMPUTER SCIENCE

CSCI 4177/5709 — Advanced Topics in Web Development

Assignments 1

Overview

In CSCI 5709, 30% of your grade involves work done for assignments. These assignments are meant to put the skills and theory you have learned in lectures and tutorials, along with the skills you had prior to joining the course, to practice. Most assignments may be considered to be individual deliverables that can be used towards your group project, although together they do not make up 100% of your group project. As such, it is strongly recommended for you to manage your time appropriately and use the assignments component of this course as an outlet for you to try out ideas you may be interested on using for your project. Furthermore, while **assignments** can be used towards your group project, they are meant to reflect the students' individual work, and therefore **are NOT to be carried out in groups**; though you may consult with group project members or classmates and TAs during lab/tutorial sessions.

There are a total of THREE (3) assignments in this course. Although, initially the assignments are not too difficult, they do get progressively harder as you learn new concepts and techniques covered in the course. As such, do keep this in mind when managing your time. **Assignments are due by the END OF DAY (i.e., 11:59PM) on the date noted on each individual handout, and must be submitted through both Brightspace and Git Lab unless otherwise specified on the assignment handout**. Finally, students should also be aware that they will be tested on topics included in each of the THREE (3) assignments, in addition to material covered in the course lectures, tutorials, and in-class discussions and activities.

It also goes without saying that any instance of **academic dishonesty** will be reported. **If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author attribution in your code comments, along with a README.txt file providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified.**

Descriptions of the assignments are posted in advance so that you are aware of what is expected in each assignment, and are able to manage your time appropriately as **assignment due dates will NOT have any extensions. You are NOT expected to submit all assignments at the same time – each assignment has its own due date**.

Any late changes (if necessary) made to this document or any of the assignments will be notified in class and via email.

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.

Assignment 1

[10% Individual Deliverable]

Assignment 1 involves the **application of User eXperience, usability and design principles** discussed in class, for the creation of a proposed Lo-Fidelity prototype of your group project's application, illustrating your ideas for the **the overall look-and-feel of a web application's UI, i.e., your group project.** Though you are encouraged to meet with your group members, keep in mind that this assignment is not a group assignment. Instead, meet with your group to discuss project ideas (e.g., purpose, features, target users) that you may use to develop your own prototype of your group's application.

The goal of this assignment is to help you draft a proposed set of requirements for your application, as well as assess the suitability of Front-End APIs for your project, given those proposed requirements, while allowing your group to take part in a **parallel design exercise** where each of your project group's members will be submitting their UI vision for the group's web application. This approach is commonly used in start-ups and design/development firms, as it often results in better team collaboration and more efficient development and integration. Remember, **each group member is expected to submit a different prototype**.

As part of this assignment, you will also have to provide justifications for the design choices you have made, e.g., APIs, Front-End Frameworks, colour scheme, typography. Finally, it goes without saying that any instance of **academic dishonesty** will be reported.

If you decide to use and modify any existing code, e.g., code found on online or printed sources, or code used during in-class/tutorial examples, you are expected to provide author attribution in your README.txt file providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified. You are encouraged to use the README template available on Brightspace, as this is an annotated template meant to guide you in this process. Keep in mind that simply stating "code was modified" does not provide sufficient information required in your programming assignments, the amount of detail expected in your README file is illustrated in the README template.

Learning Objectives:

1. Judge and apply UI and UX design techniques discussed in class (e.g., use cases, task flow diagrams), while considering the usability of the device used to access your website.

2. Assess the suitability of Front-End APIs and Frameworks for the purpose of developing a high-fidelity prototype application, given a set of proposed guidelines (e.g., wireframes, devices, expected functionality).

3. Become familiar with your group's Front-End Framework(s) of choice for the purpose of developing a semi-functioning high-fidelity prototype, given a set of proposed guidelines.

4. Learn to collaborate with group members to define your project's purpose, potential features, user personas, scenarios and sitemaps in order to compile a set of potential guidelines and requirements for your group project.

Requirements:

Part of your Assignment 1 will require for you to work with your group, and part of it will involve individual work done on your own.

For the group work portion of your Assignment 1, you must do the following:

A1.1. Project's Purpose, Goals and Intended Features

Meet with your group and define your project's proposed requirements, such as purpose, goals, its intended features (e.g., Profile Management, Permission Management, File Transfer System, Recommender System, Shopping Cart, File Management System), and your target user base.

Note: Think of this assignment as a working draft, a brainstorming opportunity. As a general guideline, we expect the number of defined features for your project to be equal to 'Group Members x 2' (i.e., if your group is made up of 5 members, you are expected to have 10 features). You will be expected to have ~70% of these features developed by your final report.

A1.2. User Personas and Intended Scenarios

Meet with your group and identify/define the user persona(s) for which you are developing your application. For each user persona, describe their characteristics, and your intended scenario(s). Finally, define the scenarios in which you envision your application being used, ensure your scenarios describe all of the required components usually found in a scenario.

Note: You are expected to provide a full description of your user personas, as shown in our lectures, simply stating "students" or "professionals" will not meet this requirement. You are encouraged to refer to the corresponding lecture materials (e.g., lecture videos and slides) for guidance in defining user personas and scenarios. Failure to provide a proper user persona description will result in a possible maximum grade of 70%.

A1.3. Sitemapping

Meet with your group to create a complete proposed sitemap for your idealized application.

Note: Your sitemap is meant to be developed as a group. Your sitemap must illustrate the complete information structure of your application, as well as areas were authentication is required. Your sitemap must be properly created using a sitemaping tool. All images in your assignment must be properly captioned and referenced within the text. You must include the site mapping tool in your References Section.

A1.4. Use Cases

Work with your group and, have each member of your group, choose at least ONE (1) of the features you defined in A1.1. Define the use cases for each task within the feature(s) you have chosen. Your use cases must clearly define the normal and alternate flow of events, you may use any of the scenarios defined in A1.2 to help you define your use case sequence.

Note: For example, if your scenario is "*it is 2am, you are in downtown Halifax and need to call a cab using our application in order to get home***", then your use case would detail the steps required (from a user and system perspective) to call a cab for this particular purpose with the environmental factors expected in that scenario. You may use bullet-point form for this item. It may be wise to draft a list of features with your group mates and assign two features to each member.**

For the individual work portion of your Assignment 1, you must do the following:

A1.5. User Experience and Task Flow

Work alone and choose ONE (1) of the features you defined in A1.1, along with their corresponding use cases from A1.4, to create a task flow diagram for each of the tasks in the feature defined in A1.1. You may submit this task flow diagram as a screenshot (e.g., .png, .jpg, .pdf).

Note: Your task flow diagrams must must be properly created using a diagramming tool, you must properly reference this tool in your deliverable. All images in your assignment must be properly captioned and referenced within the text.

A1.6. Lo-Fidelity Prototype

Choose ONE task from the feature you defined in A1.5. **Work alone** to create a lo-fidelity prototype, i.e., a wireframe, of the pages necessary for the task you have chosen, while taking into consideration the specifications you have identified in **A1.1, A1.2, A1.3, A1.4** and **A1.5**. **EACH member of your group is expected to submit a different prototype and/or design.** You may submit this task flow diagram as a screenshot (e.g., .png, .jpg, .pdf).

Note: Your lo-fidelity prototype must be properly created using a prototyping or wireframming tool, you must properly reference this tool in your deliverable. All images you may use in your assignment must also be properly captioned and referenced within the text. You must create a lo-fidelity prototype that reflects the structure and layout (at minimum) of the task and/or feature a given user persona is able to complete through your chosen page. Ensure you properly caption your wireframes (e.g., *Figure 2. Wireframe applicable to Sign Up page*).

A1.7. Semi-Functional Prototype

Work alone to create a semi-functional prototype of the pages (for the task) you chose in **A1.6.** You may use any front-end framework or languages of your choosing, but do make sure you document your work on your **README.txt** or **README.md** file. Your prototype is only meant to be semi-functional is regard to its front-end, therefore, **feel free to hard-code or use dummy data were you see fit to simulate any back-end processes.**

Note: You may discuss with group members your plans for your A1 submission, but you are asked NOT to show your actual work in order to ensure that your design does not influence other designs in your group. **Keep in mind that EVERY member of your group is expected to submit an entirely different design than yours, though some commonalities (e.g., basic wireframe structure) may be OK.**

You are encouraged to consider the use of front-end APIs where you see fit. Further, if you choose to develop a page that includes web forms, you are expected to implement front-end validation techniques that improve the usability of your prototype.

Finally, though you are tasked with creating ONE (1) page, you are encouraged to create some or all of the pages involved in a particular task, e.g., if you were to choose the registration page, then your assignment will be expected to include all pages and/or modal boxes needed to allow

a new user to complete the registration process. As such, it is possible that your assignment may include more than one page.

A1.8. The pages you develop must reflect the requirements specified by you in 1.1 through 1.6.

A1.9. You **may use *Lorem Ipsum* text** for the content of your pages. Additionally, any forms you include your design must use meaningful labels and messages (e.g., *'Your message was successfully submitted'*).

Note: Though you **may** use *Lorem Ipsum* text to help you define the content hierarchy of your submission, it is recommended for you to include meaningful text where possible as it will help you see how your design may compliment the message you are looking to communicate (e.g., headings, navigation links).

A1.10.Your assignment MUST be responsive. However, it is up to you to define the level of responsiveness your assignment should reflect, based on the requirements you specified in this assignment.

A1.11.Your assignment MUST be W3C compliant, i.e., it must pass W3C front-end validations tests (e.g., HTML and CSS).

Note: Failure to submit valid code will result it a possible maximum grade of 50%. **If your assignment does not validate due to framework-specific tags or code, these errors will be overlooked** (e.g., Angular's ng-app HTML attribute) and **WILL NOT affect your grade**. As well, **any validation warnings WILL NOT affect your grade.**

A1.12.Your assignment MUST apply usable front-end validation and user feedback techniques to validate form fields, and provide proper error recovery messages in case a field does not validate.

Note: Proper user feedback in forms may include the use of AJAX confirmation or success messages, as well as failure messages to the user. Your messages should also take into consideration the security of your application. Of course, what you implement is based on your vision for your project.

A1.13. In regards to the look-and-feel of your assignment, you have complete creative freedom for this assignment. You are encouraged to work towards an aesthetically pleasing website that applies the design and development principles discussed in class. You may use Creative Commons images and/or logos with proper author attribution (provided through code comments, and/or **README.txt** file).

Note: Do keep in mind that as part of this assignment, you are expected to work individually on a specific design. You may, if agreed by your group, use the same HTML structure for your A1 submission. However, you **CANNOT** 'share' any CSS code.

A1.14. Make sure to **include in your README.txt file,** the URL from which your assignment can be accessed. All pages you develop for this assignment will need to be accessible through that link, otherwise, you may include the links to all individual pages.

Note: If you decide to use and modify any existing code, e.g., code found on online or printed sources or code used during in-class/tutorial examples, you are expected to provide author

attribution in your code comments, along with a README.txt file providing an explanation of why the piece of code is necessary for your work, where, how and why the code or section of code was modified. Keep in mind that simply stating "code was modified" does not provide sufficient information required in your programming assignments.

Dimensions	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
Formal Writing (5%)	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, thingy, stuff). (1 points)	Uses mostly a formal writing style with minimal use of slang (i.e., < 6) or abbreviations. (3 - 4 points)	Uses formal writing style with **no** use of slang or abbreviations. (5 points)
References (5%)	Fails to reference sources using in-text citations, or does not use proper in-text citations (e.g., instead uses "In the first article"). Inconsistent citation style (e.g., sources are in IEEE and ACM in the document). (1 points)	A single citation style is used consistently with minimal errors (i.e., < 6). Most sources are referenced throughout the text with few missing in-text citations (i.e., < 6). Most sources correctly included in the References section. (3 - 4 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. (5 points)
Grammar (5%)	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 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). (3 - 4 points)	Great grammar and sentence structure. Paragraphs are well structured. Document is easy to navigate and read through (i.e.,< 1 mistakes). (5 points)
Content (15%)	Excessive lack of detail leading to vague sentences. Content is hard to follow due to missing details. Figures not correctly captioned and/or referenced within the text (e.g., 'As shown on Figure 2,…'). (1 - 5 points)	Some vague sentences and missing details. It is relatively possible to follow the content despite missing details. Most figures correctly captioned and referenced. (8 - 10 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. (12 - 15 points)
Completeness (15%)	Sections left blank. Paragraphs/sentences end midway (i.e., incomplete). Did not follow the template provided in class for the deliverable. Writer does not clearly state the project's purpose, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas. The reader is not referred to any Figures and/or they do not have a proper description provided within the text. (1 - 5 points)	Sections seem to be mostly complete. Mostly followed the template provided in class for the deliverable. Writer somewhat states the overall project purpose/goals, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas. The reader is referred to some Figures and/or some do not have a proper description provided within the text. (8 - 10 points)	All sections completed, used the template provided in class for the deliverable. Project purpose/goals, target user base, scenarios, use cases, task flows, sitemap, prototype, user personas are clearly stated, meeting the expectations of the deliverable. The reader is referred to ALL Figures and ALL Figures have a proper description provided. (12 - 15 points)
Clarity (5%)	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 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. (5 points)

Marking Rubric

The following grading criteria, shown in Table 1, will be used for marking the group work portion of your as-

signment:

The following grading criteria, shown in Table 2, will be used for the individual work portion of your assign-

ment:

Dimensions	Does Not Meet Expectations	Meets Expectations	Exceeds Expectations
Web Pages (5%)	Fails to meet the criteria of the pages necessary to complete at least ONE (1) task. (0 points)	Meets the criteria of the pages necessary to complete at least ONE (1) task, but pages seem incomplete or not user cannot complete the task with the pages submitted. (3 - 4 points)	Assignment submission meets or exceeds the expectation of the pages necessary to complete at least ONE (1) task. (8 - 10 points)
Front-End Frameworks (10%)	Fails to implement front-end frameworks and/or does not provide justification for not using a framework. (1 - 3 points)	Implements front-end frameworks but fails to customize its implementation. (5 points)	Successfully implements and customizes front-end frameworks. (8 - 10 points)
Responsiveness (5%)	Fails to successfully implement a responsive layout (i.e., does not implement it or fails to be cross-browser compatible). (1 - 2 points)		Successfully implements a responsive layout that is W3C compliant and cross-browser compatible. (5 points)
Content (5%)	Fails to implement proper content hierarchy throughout. Content is not well organized. Poor use of Lorem Ipsum content. (1 - 2 points)	Content hierarchy is somewhat defined. Content organization is still somewhat lacking. (3 - 4 points)	Successfully implements proper content hierarchy throughout. Content is well organized. (5 points)
Design (10%)	Overall design is cluttered and fails to reflect the requirements set out in Assignment 1. Assignment does not have a clear colour palette or typographic style defined. Design elements are not consistent. (1 - 2 points)	Overall design is somewhat cluttered but does seem to reflect the requirements set out in Assignment 1. Assignment seems to have a clear colour palette but no clear typographic style defined. Design elements are somewhat consistent. (4 - 6 points)	Overall design is clean, aesthetically pleasing and successfully reflects the requirements set out in Assignment 1. Assignment has a clear colour palette and typographic style defined. Design elements are consistent and enhance the usability of the application. (8 - 10 points)
W3C Compliant (5%)	Fails to be cross-browser compliant and/or implement W3C valid code and/or displays cross-browser compatibility issues. (-25 points)		Assignment is cross-browser compliant and/or implements W3C valid code and does not show any cross-browser compatibility issues. (0 points)

UX and Usability (5%)	Fails to implement usable front-end validation and/or user feedback techniques throughout. Assignment has a lack of error recovery messages. (0 points)	Implements some front-end validation and/or user feedback techniques throughout but messages used are not usable or do not provide a clear idea to the user of the issue or how to correct it. (1 - 2 points)	Assignment properly applies usable front-end validation and/or user feedback techniques throughout, with proper error recovery messages. (5 points)
README.txt (5%)	Fails to include a README.txt file, or file is empty. Fails to include details on how a given block of code was required for the assignment, and/or how it was modified. (0 points)	Includes a README.txt file but any instructions or content Is incomplete or incorrect. Code referencing lacks information. (1 points)	Includes a README.txt file with complete and correct content. Code referencing, where needed, is done correctly and includes expected details. (5 points)

Submission Guidelines

Your assignment must be submitted through **Brightspace**.

To submit your work to Brightspace:

- **For your group submission**, include your answers to A1.1, A1.2, A1.3, and A1.4 (which is work you've done with your group), as well as A1.5 and A1.6 in a single **PDF** file, these answers will be a part of your **group submission**. Your group submission must match naming conventions specified in the Course Syllabus. In this case, since these sections represent work done in groups (with the exception of A1.5 and A1.6), your written portion of this assignment should be named **A1_Group#.pdf and be submitted into the A1 Group Submission assignment dropbox on Brightspace.**

  Note: Any deliverable not submitted as a PDF file will have a 5% grade deduction. Any deliverable submitted without following the proper file naming convention will have an additional 5% grade deduction. **Include the name of the group members who worked on a particular task flow diagram (i.e., A1.5) and Lo-Fi Prototype (i.e., A1.6) within their corresponding Figure captions.**

- **For your individual submission,** include your answers for A1.5 and A1.6, and README.txt or README.md in a **compressed ZIP file**. Your README file must provide the appropriate details for the technologies used and work done in developing your individual semi-functional prototype, as well as the prototype's URL, and must match naming conventions specified in the Course Syllabus, i.e., it must be named **A1_README_LastName_FirstName.txt** or **A1_README_LastName_FirstName.md**, **and be submitted into the A1 Individual Submission assignment dropbox on Brightspace.**

  Note: Your compressed zip file for your individual submission must be named **A1_LastName_FirstName.zip**

To submit your work to Git Lab:

- Within your GitLab main repo for this course (i.e., 'csci5709' or 'csci4177'), push your code to a new branch (e.g., 'assignment1') and deploy the application to Heroku or any other deployment platform of your choice.

Note: Make sure that the deployment link you include in your README file matches the deployment link for this assignment. Your GIT repository must be individual and private and be accessible to the Instructor and Teaching Assistants, see Tutorial 2 handout for Instructor and TAs GitLab usernames.

- 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 will be provided in class/tutorials. Failure to add the course CS ID as 'Maintainer' for your work on Git Lab will result in a maximum possible grade of 50%.

- You are free to use GitHub if you wish to do so instead of GitLab, but you must ensure you provide access as '**collaborators**' to the Markers provided in your **Tutorial 2 Handout**.

- **Create a README.txt or README.md file**, follow the guidelines specified in the README template provided through Brightspace, and naming conventions specified in the Course Syllabus.

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 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, for the sake of simplicity, create a folder called **'csci5709'**.

  Note: All your work **must** be reside inside your '**csci5709**' 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 **'csci5709'** folder and create an assignment folder called **'a1'**.

  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., A1, a3, a4, and project).

- Place the all the files you created for this assignment inside the **'a1'** 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 .

- 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 **A1** such as those that can be achieved through the use of CSS and Javascript. **However, bear in mind the following:**
  - Your submission **must** meet the criteria specified in A1, 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 **A1.**
  - You stand to lose points if the additional markup / CSS elements that you might implement interferes with the basic requirements of **A1.**
  - 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.

Academic Integrity[1]

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?

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

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.
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.