**Web Application Development**

Project Specifications

Overview

Web technologies and applications are essential for enabling [e-commerce](http://digitalenterprise.org/models/models.html)—the buying and selling of goods and services over the Internet. E-commerce has both given rise to new kinds of business models and reinvented tried-and-true models. In this project, you will propose and develop a web application that promotes the selling or use of a service or product.

A typical e-commerce web application involves five major activities: information sharing, ordering, payment, order fulfillment, and customer service and support.

* Your web application should at least include the information sharing activity, which consists of using web technologies to share information with potential customers or users of your application.
* Your application should not include payment or order fulfillment activities, which means you should not implement any form of payment processing.

What will you do?

This document contains the requirements of your web application. You will complete the requirements in three iterations:

1. Iteration #1: Business Case (proposal)
2. Iteration #2: Front-end Implementation
3. Iteration #3: Complete Implementation

Visit the Canvas assignments for each iteration to learn about the deliveries and due dates of each iteration. The purpose of developing your project in several iterations is to receive feedback from your instructor during your application development as you would if developing your project in the industry.

Read ALL the instructions listed on this document BEFORE start working on the first iteration.

## Iteration #1: Business case (proposal)

In the real world, to reap the benefits of e-commerce, companies often invest large amounts of resources in developing web applications. Before starting developing your application, you need to write a business case to explain the purpose of your application from a business perspective and justify its development.

When writing the business case for your application, you may come up with a fictitious company and think about the products or services your application will promote. You might do some research and take a look at sites or applications offering similar products or services. Nonetheless, you must reflect and elaborate on the novelty and originality of your application. Use your creativity to the fullest to define the character and purpose of your application that would make it attractive and interesting to an audience.

Your business case should not exceed more than one page in a font size no smaller than 10 points.

* Follow the template provided in the [Project-business case template](https://docs.google.com/document/d/1t9JiOgVFgOZMt3pFFU8mOYTZjU7EfVQHw6m9RZ2Ebv8/edit?usp=sharing) document.
* As an example, take a look at the [business case discussed in class](https://docs.google.com/document/d/1mm4LbBwQdFnHor9dPLgdEQX8Hf-NU41yeoELF1Lu_zU/edit?usp=sharing), and the guidelines for designing the content of websites and web applications also discussed in class in Week 2.

## Iteration #2: Front-end implementation

In this iteration, use HTML, CSS, and JavaScript to implement the front-end of your application. When doing so, remember the following points discussed in class:

* Follow the principles of visual hierarchy and grouping and similarity
* Implement as many on-page SEO techniques as possible
* Make your web application responsive, i.e., your application should render well on any window or screen size.

You are responsible for finding and writing all the content and resources during the development of your application. You may use [Google Images](https://www.google.com/imghp) and stock photos sites like [Free Images](https://www.freeimages.com/) to look for pictures. When using Google Images, you must be aware of copyrights and make sure you are using [free-to-use](https://support.google.com/websearch/answer/29508?hl=en) images.

### Add user interactivity

Apply your knowledge in JavaScript to implement features that enhance the interactivity of your application. Aim at creating a richer user experience by enabling the user to be actively engaged. You may implement components, such as:

* **Search bar**. A search bar that allows users to search for items or other information in your application. When implementing this feature, you must display the resulting items dynamically from the search.
* **Collapsible content**. An option that allows the user to display more information about the items sold by your application. For example, in the content shown for each item, there could be a text element enabling the user to see or learn more about the item's description when the text is clicked. Once the expanded description appears, the text could change to a "show less" message and hide the description when the text is clicked.
* **Filtered content**. You can help users find the information they need by providing filters. You can have buttons or text so that when the user clicks on one of these elements, they are only shown the content corresponding to the element.
* **Slideshow**. Slideshows can display several different images (or other HTML content) within the same space on a given page. They can play automatically as a sequence, or users can click through the slides manually. They allow more content to be displayed within a limited amount of space.
* **Shopping car**. Allows visitors to select items for eventual purchase.
* Any other feature that allows the user to engage with the application actively.

### Use version control

Keep track of your changes using Git and post your commits to a GitHub repository. After finishing your application, the commit history of your GitHub repository should reflect the development process of your application. Therefore, you should be mindful of writing descriptive messages in your commits.

### Publish the front-end of your application online

Finally, get the front-end of your application online using GitHub Pages.

## Iteration #3: Complete implementation

In this iteration, complete your implementation by adding back-end functionality using Python and Django. You may implement functionality, such as:

* **Make your content dynamic.** Improve the management and maintenance of your application by enabling data management on the back-end side. You are free to choose the data handled in the back-end and the mechanisms for data management and including the data in the front-end.
* **Add data persistence**. Improve user experience by capturing and storing information provided by the users and visitors of your application. You are free to choose the mechanisms for requesting the data and storing it on the back-end.

How do you submit your Project?

Visit the Canvas assignments for each iteration to learn about the deliveries and due dates of each iteration.

Criteria for Success

* An excellent application is not the one with the highest number of code lines. You may compensate for the lack of advanced programming skills with a solid and creative business case.
* Use Grammarly to revise the grammar and structure of your writing, especially when writing the business case.
* You are allowed to use the code given to you in tutorials and lab practices throughout the semester.
* You are not allowed to use any website builder. You must implement your application by writing code in HTML, CSS, JavaScript, and Python. Although high-level website builders are becoming increasingly popular (e.g., Wix), learning the core Web technologies can give you significant advantages in your career even if you do not want to become a web developer.
* Bear in mind that if you are getting images from other sites, you must be mindful of copyrights and make sure you are using [free-to-use](https://support.google.com/websearch/answer/29508?hl=en) images.