

Umm Al-Qura University  
College of Computing Sciences and Engineering  
Department of Information Science

**INFT 2202**

**Assignment 1**  
**DOM Manipulation**

# Table of Contents

## Contents

INFT 2202 - Assignment 1 .....	3
Assignment Objective: Interactive Community Center Web Portal .....	3
Theme Description: .....	3
Application Goals: .....	3
Assignment Instructions.....	4
Project Setup .....	4
Project Content.....	4
Interactive Feature and DOM Manipulation .....	5
Internal Documentation .....	5
GitHub Share .....	5
Host your site .....	6
Video Presentation .....	6
Submission.....	6

# INFT 2202 - Assignment 1

## Assignment Objective: Interactive Community Center Web Portal

This assignment is designed to immerse you in the practical world of client-side scripting, focusing on essential skills like DOM manipulation, event handling, and dynamic content generation. Your task is to develop a web application for a fictional community center named "**Harmony Hub**."

### Theme Description:

The Harmony Hub is a vibrant, community center that aims to offer a range of services and activities to people of all ages and backgrounds. It's a place where technology meets community service, providing resources like classes, events, workshops, and community projects. Your web application will serve as an interactive portal for the Harmony Hub, showcasing its offerings, engaging the community, and facilitating online interactions.

### Application Goals:

- To create a dynamic, user-friendly website that reflects the spirit of Harmony Hub.
- To demonstrate real-time user interaction through web technologies.
- To build a scalable web application that can evolve and incorporate new features in future course iterations.

As you develop this web application, consider how it can be structured to accommodate future expansions like **member sign-ups**, **event registrations**, **interactive forums**, or even a **small e-commerce module** for community-made products. These may be enhancement that are asked of you in future assignment in this course. This project not only tests your technical skills but also challenges your creativity in designing a web interface that is both inviting and functional for a diverse group of users.

Your website should embody the essence of the community center: a welcoming, engaging, and resourceful online space that mirrors the vibrancy and diversity of the community it serves. Use this opportunity to experiment with different layouts, interactive elements, and content that bring the community center to life in the digital realm.

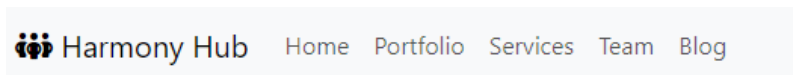
By the end of this course, you should be able to present a comprehensive web application that not only demonstrates your technical proficiency in client-side scripting but also your ability to conceptualize and execute a thematic project that could be realistically implemented in a community setting.

This assignment, therefore, is not just about coding; it's about crafting a digital experience that resonates with the users and serves a meaningful purpose. Let your imagination and technical skills come together to create something unique for the Harmony Hub.

## Assignment Instructions

### Project Setup

1. **Folder Structure:** Set up your folder structure as demonstrated in class, ensuring clean separation of HTML, CSS, and JavaScript files.
2. **Basic HTML Page:** Create a basic (index.html) HTML page. The page must link to your **style.css** and **main.js** files stored in designated folders. Include libraries like jQuery, Bootstrap, and a new library of your choice (e.g., **Axios**, **Lodash**) using the **yarn** package manager. The index.html file should be placed in the root directory of your project folder. Inside your index.html, link the style.css and main.js files. These files are typically stored in folders named css and js, respectively as demonstrated in class.
3. **Bootstrap Navigation Bar:** Include a navigation bar with the following links: **Home**, **Portfolio**, **Services**, **Team**, and **Blog**. Ensure the use of **Bootstrap** for styling.



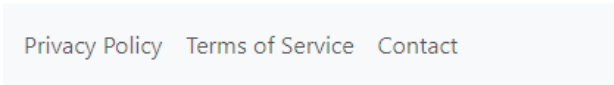
### Project Content

- i. **Home Page:** Design an engaging home page with a dynamic background (e.g., a slideshow of images or a video background) and welcoming text.
- ii. **Portfolio Page:** Display a list of your (hypothetical) projects. Use JavaScript to dynamically create project cards that include a **title**, a **short description**, and an **image** for each project. Implement a '**Load More**' button to add more projects to the page without reloading.
- iii. **Services Page:** Describe three services you offer, using icons or images for visual representation. Implement an interactive feature, like a tab or accordion layout, to view the details of each service.
- iv. **Team Page:** Present your team members using a grid layout. Each member should have a photo, name, and a brief role description. Implement a modal popup that shows more details about each member when clicked.
- v. **Blog Page:** Create a simple blog page displaying articles with titles, summaries, and read more links.
- vi. **Privacy Policy Page:** Develop a Privacy Policy page that outlines how user data is handled. This should include sections on the types of data collected, usage of the data, and user rights. Ensure the content is clearly structured and easy to navigate.
- vii. **Terms of Service Page:** Create a Terms of Service page detailing the rules and guidelines for using the website. Include sections on user responsibilities, copyright information, and disclaimers. The page should be well-organized and user-friendly.
- viii. **Contact Page:** Construct a Contact page featuring a form for users to submit their queries or feedback. The form should include fields for the user's name, email, subject, and message. Implement form validation to ensure data integrity and provide a user-friendly experience.

Additionally, include your (or the fictional entity's) contact information, such as an email address, phone number, and physical address.

## Interactive Feature and DOM Manipulation

- i. **Dynamic Navigation Bar:** Using JavaScript, dynamically add a '**Careers**' link to the navbar. Also, programmatically change the '**Blog**' link to '**News**'.
- ii. **Event Handling:** Implement a form on the **Contact** page. On submission, validate the input fields, display the entered data in a modal, and reset the form. Prevent default submission behavior.
- iii. **Timed Redirection:** After form submission, display a thank you message with a countdown timer of 5 seconds, after which the user is redirected to the Home page. Please research how to do this. I recommend the following resource ([https://www.w3schools.com/js/js\\_timing.asp](https://www.w3schools.com/js/js_timing.asp)) - **setTimeout(...)**
- iv. **Additional Navigation Bar:** Using JavaScript, add a footer navigation bar fixed at the bottom of the page with links to your **privacy policy**, **terms of service**, and **Contact page**.



Privacy Policy   Terms of Service   Contact

## Internal Documentation

- **Header Comments:** Include a comment header in your main.js file with your name, student ID, and date of completion.
- **Function Documentation:** Document all functions with clear descriptions, parameter details, and return values.
- **Readable Code:** Use meaningful variable and function names and include inline comments where necessary to clarify your code.
- **Responsive Design:** Ensure your web application is responsive and looks good on both desktop and mobile devices.

## GitHub Share

- a. Share your repository with your professor (collaborator) on GitHub. Create an appropriately named **private** repository that you and your partner(s) will use for this assignment. Only 1 repository is required per submission.
- b. Your repository must include your well-structured code.

## Host your site

- a. Host your site live on a Cloud Provider of your Choice (GitHub pages recommended)
- b. Your site's images should be visible on your Live site
- c. Your CSS Layout should function appropriately on your Live Site
- d. Your Script files should function appropriately on your Live Site

## Video Presentation

- a. Create a Short Video presentation. Your presentation should start with an introduction, where it must include a PowerPoint or Google slide that includes 1 single slide to, that introduces each member of your group, to start your video.
- b. The first (and only) slide of your presentation must include current images of you and your partner(s) (no avatars allowed) that are displayed appropriately on the page. You must also include your Full Names, Student IDs, the Course Code, Course Name, and your Assignment information.
- c. You or your partner(s) will demonstrate your program's functionality. You must show your site working properly on your live site. You will also use your assignment status report as a checklist during your Video Presentation
- d. You or your partner(s) will each share in describing the code in your files that drives the functionality of your program – you will want to do this part well and be clear.
- e. Sound for your video must at an appropriate level so that your voices may be clearly heard, and your screen resolution should be set so that your program's code and console details are clearly visible
- f. Your Short Video should run no more than ~5 minutes

## Submission

1. A zip archive of your project uploaded to DCConnect
  - a. Please include your video recording (mp4 preferred)
2. A working link to your completed project files on GitHub
3. A working link to your Live Site hosted on a Cloud Provider of your Choice (GitHub pages recommended)
4. Your completed Assignment Status Report
5. Late submissions are deducted **-25%** per day (max 3 days)