**College of Engineering**
**Computer Science & Eng. Dept.**
**Course:** COE 457 Internet and IoT Programming (Lab)
**Date**: 22nd September 2020
**Location**: Online

**Course Instructor:** Dr. Imran Zualkernan
**Email**: izualkernan@aus.edu

**Lab Instructor**: Ms Hend ElGhazaly
**Emails**: helghazaly@aus.edu

# Lab 4: HTML and Curl

**Objectives:**

- To create static forms in HTML5.
- To use Curl to connect to servers and view HTTP headers.
- To understand how the browser handles GET and POST request.

***Hand in:*** ***One team member needs to*** upload on iLearn:

- The solution document with screenshots of output.
- The HTML and CSS files.

***Due Date:*** Wednesday 23rd September, 11:59pm (5% per day will be applied on late submissions).

- **Useful resources:**
  - **Lectures:** The HTTP Protocol and HTML
  - **GitHub Repository:** https://github.com/izualkernanaus/COE457fb
  - **W3Schools HTML tutorial**: https://www.w3schools.com/html/default.asp
  - **Curl Options:** https://gist.github.com/subfuzion/08c5d85437d5d4f00e58

## Exercise 1: Smart lab dashboard web page

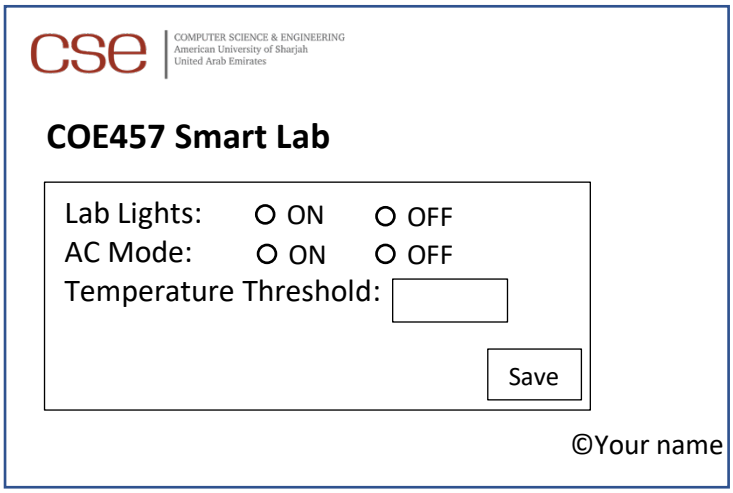
Create an HTML page that has an image and a form with options to switch on/off the appliances and set the temperature threshold of the lab.

The webpage should have:

- An image of the CSE logo on the left
- The image should be clickable and links to the AUS CSE website (http://auscse.com/)
- A header with the heading: COE457 Smart Lab
- A form with
  - Radio buttons with on/off options
  - Text field for temperature threshold
  - A Save button.

  *Note:* You should use the POST method in the form.
- Footer element with your name

The page should look similar to this:




## Exercise 2: Sending a GET/POST request to a server

a. Modify the server from the HTML lecture[1] (server_http_wf.js) to make the server send back the contents of your HTML file containing the form you created in Exercise 1.
b. Browse to *localhost:8080* in the browser to fetch and submit the form. Show the console output from the server.
c. Use the **curl** to connect to the modified server and to simulate how a form works in a browser:
   a. Do a GET to fetch the HTML form
   b. Do a POST with appropriately filled fields and body using:
      i. URL encoded

```
curl –d “@data.txt” –X POST http://localhost:8080
```

      ii. JSON (sending a JSON file)

```
curl –d “@data.json” –X POST http://localhost:8080
```

[1] https://github.com/izualkernanaus/COE457fb