F21DV Coursework Part 1 Report

Jonathan Song Yang, Lee

H00255553

Contents

1 Introduction

The entirety of the F21DV four-part coursework is done in a way where it resembels a full static or a `node.js` server-side web application. The goal of this series of coursework is to demonstrate the understanding of `d3.js`.

1.1 General Setup

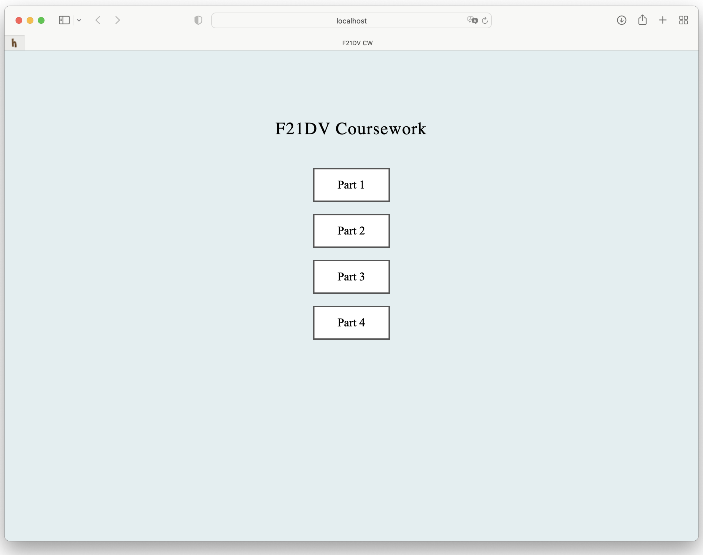
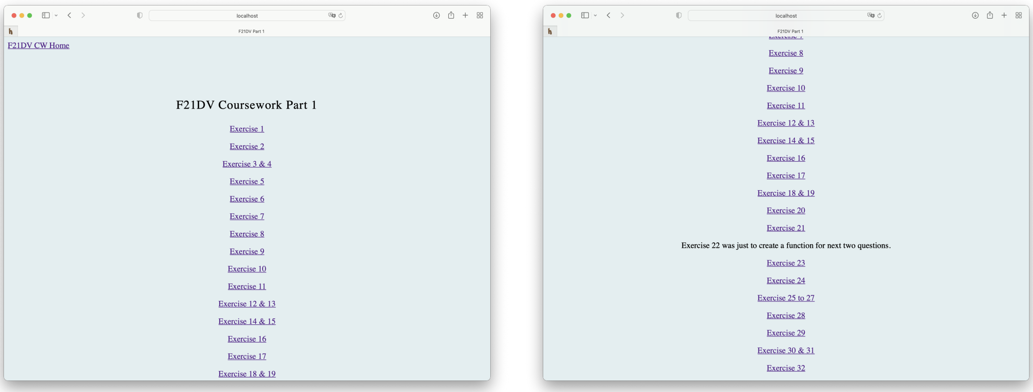


Figure 1.1: Main Index Page

The main index page of the web application would show buttons to access parts 1 to 4, as shown in figure 1.1, and for the case of Lab 1, upon clicking on the Part 1 button, a series of urls linking to different exercises would be shown, as seen in figure 1.1 below.



Exercise 22 was just to create a function for next two questions.

Figure 1.2: Main Index Page

To demonstrate the understanding of `d3.js`, this series of URLs were created systematically using `d3.js`, as shown in listing 1.1 abstact below.

```
// Array of numbers [0, 1, 2, ..., 32] defining number of exercises
const data = Array.from(Array(33).keys());

// Creating links to 32 Exercises Systematically.
d3.select('body').selectAll('p')
    .data(data)
    .enter()
        // Append a <p> for each exercise and add an <a> for each <p>
        .append('p').style('text-align', 'center')
            .attr('class', d => 'task' + d)
            .append('a')
                .attr('href', d => 'task' + d +'.html')
                .html(d => 'Exercise ' + d);
```

Listing 1.1: Systematic URL Creation

As the lab exercise would involve certain combinable exercises, such as exercises 30 and 31, I have also created a generalised function to help **combine** and **delete** certain exercises.

```
/**
* General Merge Function. If its just 2 consecutive exercise, ignore 3rd and 4th parameter.
* However, if merge is for a range of exercises, spanning more than 2, only enter the
* first and last exercise number.
* @param {*} first first exercise to merge
* @param {*} second second exercise to merge. Last exercise if there are more than 2 exercises.
* @param {*} cond1 used to change exercise <number> to Exercise <number> to <number>
* @param {*} cond2 used to rename html file to task<first>n<second>.html
*/
function mergeTask(first, second, cond1 = ' \& ', cond2 = 'n') {
    d3.select('.task' + second).remove();
    d3.select('.task' + first + ' a').html('Exercise ' + first + cond1 + second)
                                      .attr('href', d => 'task' + first + cond2 + second + '.html');
}
```

Listing 1.2: Systematic URL Removal

Listing 1.2 shows a function that renames the href of combined exercises, and removes and rename a pre-existing href. For example, we have initially Exercises 30 and 31. The function first removes the exercise 31's `<p>`, and then renames the original html href file from `task30.html` to `task30n31.html`.

```
// Remove task function
function removeTask(task, message) {
    d3.select(`.task${task} a`).remove();
    d3.select(`.task${task}`).append('p')
                            .text(message);
}
```

Listing 1.3: Systematic ¡a¿-text Replacement

Listing 1.3 shows a generalised function that replaces the exercises URL with a message string. For example, in exercise 22, we were asked to create a generalised SVG and lines function. This function will now remove the `href` from the `<p>` element, and replace it with a message string.

1.2 Individual Exercise HTMLs Setup

Within each exercise's own HTML `body`, they all inherit a generic CSS setting for a div called `.answerCenter` which acts as the center element for the presentation of answers for each exercise. There is also a generic button cssd style called `.buttonOri` and `.button` that handles the CSS transition of all the buttons.

For each exercise, I have also used a generalised function in `functions.js` to create the `<div>`s and buttons so that the styles remain consistent across all exercises.