**COMP 9783 Front-end Development**

**Lab-2-2 - CSS Frameworks Hello World. (4% of the course mark)**

Name:

Student Number:

This lab introduces students to three popular CSS frameworks: Bulma, Bootstrap, and Tailwind CSS. Through a hands-on "Hello World" project, students will learn how to set up, customize, and compare the fundamental features of each framework, including layout, typography, and styling utilities.

**Lab objectives:**

1. Understand how to integrate Bulma, Bootstrap, and Tailwind into a project via CDN or npm.
2. Create responsive layouts using grid systems or utility classes specific to each framework.
3. Apply basic styling (colors, typography, buttons, and forms) using predefined classes.
4. Explore framework-specific customization.
5. Identify the key differences between the three frameworks and their ideal use cases.
6. Assess the pros and cons of utility-first design (Tailwind) versus component-based frameworks (Bulma and Bootstrap).
7. Build and deploy a simple, responsive "Hello World" webpage using each framework.

**Create an Express.js app:**

1. On **VSCode**, **create** a **folder** named **Express-Static-CSS-FW-App**.
2. Open the **terminal** and change the directory to **Express-Static-CSS-FW-App**.
1. Initialize the app by typing **npm init -y** and **press enter**.

***Note: npm init -y is a quick shortcut to initialize the package.json file. It will initialize the app with default values.***

3. Type **npm install express** and **press enter**.
4. Type **npm install -D nodemon** and **press enter**.

***Note: The -D option means the package would be installed as a developer dependency.***

5. Open **package.json** and look for the **scripts** attribute and add this code:

"start": "nodemon index.js"

6. Save the changes.
7. Create a file named **index.js** and enter the following code:

```
const express = require("express");
const app = express();
const port = 3000;
const APP_NAME = "Express-Static-CSS-FW-App";

app.use(express.static("public"));

app.listen(port, () => {
  console.log(`${APP_NAME} listening on port ${port}`);
});
```

8. Save the changes.
9. Create the following directory structure.
    a. **Express-Static-CSS-FW-App/public**
    b. **Express-Static-CSS-FW-App/public/bulma**
    c. **Express-Static-CSS-FW-App/public/bootstrap**
    d. **Express-Static-CSS-FW-App/public/tailwind**

10. On the terminal enter the **npm run start** and **press enter**.

**Bulma:**

1. Navigate to: **Express-Static-CSS-FW-App/public/bulma** and create a file named: **index.html**.
2. Enter the following code:

```html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bulma Hello World</title>
    <link
      rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bulma@1.0.2/css/bulma.min.css"
    />
  </head>
  <body>
    <section class="section">
      <div class="container">
        <h1 class="title">Hello World</h1>
      </div>
    </section>
  </body>
</html>
```

3. Save the changes.
4. Open your browser to http://localhost:3000/bulma/ and take a screenshot and name it **bulma01.png**.
5. Navigate to: https://bulma.io/documentation/ and implement one bulma feature by modifying **index.html**. Once the bulma feature has been implemented, navigate to http://localhost:3000/bulma/ and take a screenshot and name it **bulma02.png**.

**Bootstrap:**

1. Navigate to: **Express-Static-CSS-FW-App/public/bootstrap** and create a file named: **index.html**.
2. Enter the following code:

```html
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Bootstrap Hello World</title>
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
      crossorigin="anonymous"
    />
  </head>
  <body>
    <h1>Hello World</h1>
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN7N6jIeHz"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

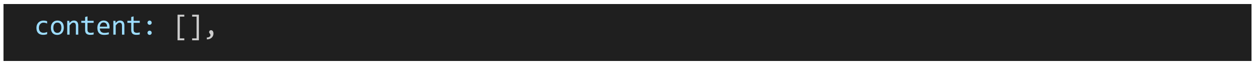
3. Save the changes.

4. Open your browser to http://localhost:3000/bootstrap/ and take a screenshot and name it **bootstrap01.png**.
5. Navigate to: https://getbootstrap.com/docs/5.3/getting-started/introduction/ and implement one bootstrap feature by modifying **index.html**. Once the bootstrap feature has been implemented, navigate to http://localhost:3000/bootstrap/ and take a screenshot and name it **bootstrap02.png**.

**Tailwind:**

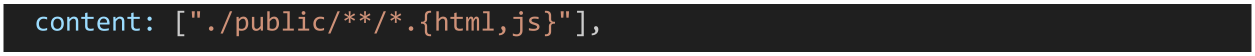
1. Open the **terminal** and change the directory to **Express-Static-CSS-FW-App**.
2. Install tailwind by typing: **npm install -D tailwindcss**.
3. Create **tailwind.config.js** by typing: **npx tailwindcss init**.
4. Open **tailwind.config.js** and update the value of **content**:

From:



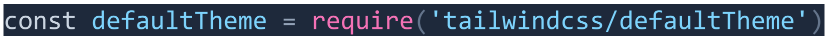
```
  content: [],
```

To:



```
  content: ["./public/**/*.{html,js}"],
```

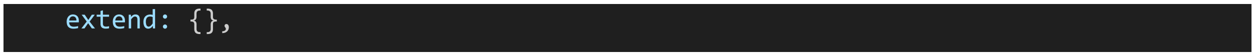
5. Add this require statement on the top of the file.



```
const defaultTheme = require('tailwindcss/defaultTheme')
```

6. Update the value of **extend**:

From:



```
    extend: {},
```

To:

```
    extend: {
      fontFamily: {
        sans: ['InterVariable', ...defaultTheme.fontFamily.sans],
      },
    },
```

7. Save the changes.
8. Navigate to: **Express-Static-CSS-FW-App/public/tailwind** and create a file named: **input.css** and enter the following code:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

9. Save the changes.
10. Navigate to **Express-Static-CSS-FW-App** and on the **terminal** type: **npx tailwindcss -i ./public/tailwind/input.css -o ./public/tailwind/output.css --watch**.
11. Navigate to: **Express-Static-CSS-FW-App/public/tailwind** and create a file named: **index.html**.
12. Enter the following code:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <title>Tailwind Hello World</title>
    <link href="./output.css" rel="stylesheet" />
  </head>

  <body>
    <h1 class="text-3xl font-bold">Hello World</h1>
  </body>
```

```
</html>
```

13. Save the changes.
14. Open your browser to http://localhost:3000/tailwind/ and take a screenshot and name it **tailwind01.png**.
15. Navigate to: https://tailwindui.com/components/preview/ and implement one tailwind feature by modifying **index.html**. Once the tailwind feature has been implemented, navigate to http://localhost:3000/tailwind/ and take a screenshot and name it **tailwind02.png**.

**Submission:**

1. Use the html template: **index.html** and write **HTML codes for each screenshot**:
   a. Write a title and short description.
   b. Display the screenshot.
2. Create a new folder named **html** and copy all the **HTML**, **CSS** and **PNG** files used in the previous step.
3. Create a new folder named **app** and **copy all** of the files from the **Express-Static-CSS-FW-App** except for the **node_modules** folder.

***Note: Do not include the node_modules folder, this is not needed as I can just run npm install to reinstall the packages.***

4. Create a new folder named **submit** and copy both the **html** and **app folders**.
5. Create a **zip file** of the **submit folder**.
6. Submit the **zip file** to **GBC - D2L**.