WEB700 Web Programming

Assignment # 4

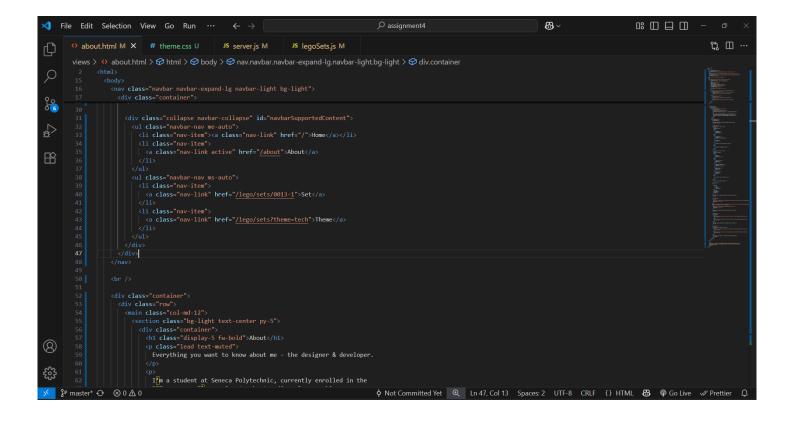
Student ID: 112867247 Student Name: Kulsum Timol E-mail: katimol@myseneca.ca

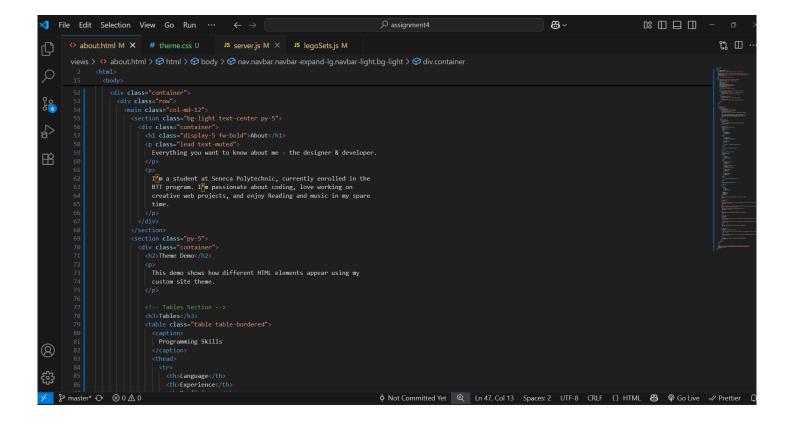
Semester:2

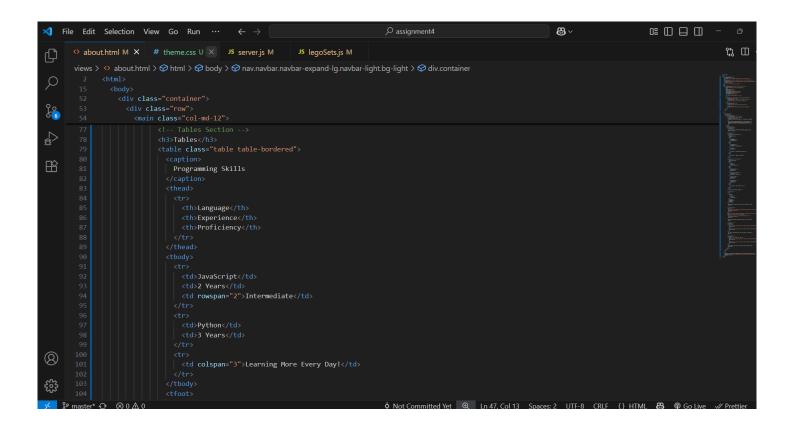
Professor name: Pritesh Patel

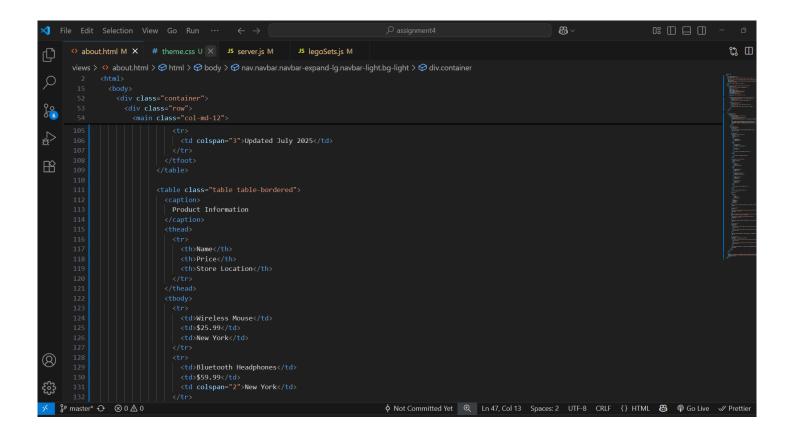
About.html File

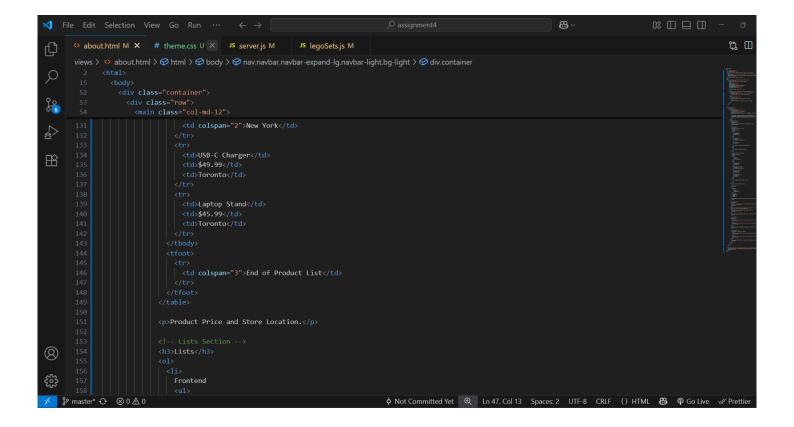
```
刘 File Edit Selection View Go Run …
                                                                                                                                                                               83 ~
                                                 JS server.js M × JS legoSets.js M
                                                                                                                                                                                                     th Ⅲ …
         views > 🥎 about.html > 🔗 html > 🥱 body > 🥱 nav.navbar.navbar-expand-lg.navbar-light.bg-light > 😭 div.container
Q
96
                    <meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
₽
                      integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
                    <a class="navbar-brand" href="#">Student Name</a>
                         data-bs-toggle="collapse"
data-bs-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent"
aria-expanded="false"
                          aria-label="Toggle navigation"
                          class="navbar-nav me-auto">class="nav-item"><a class="nav-link" href="/">Home</a>class="nav-item">class="nav-link" href="/about">About
(8)
£$$
                    ♦ katimol (3 weeks ago) € Ln 30, Col 1 Spaces: 2 UTF-8 CRLF {} HTML 🔠 © Go Live
```











```
⇒ about.html M × # theme.css U × JS server.js M
                                                                                         JS legoSets.js M
views > 💠 about.html > 🔗 html > 🔗 body > 🚱 nav.navbar.navbar-expand-lg.navbar-light.bg-light > 🔗 div.container
         <!-- Images Section -->
<h3>Images</h3>
                     src=https://legendary-digital-network-assets.s3.amazonaws.com/wp-content/uploads/2821/87/12215912/Harvel-LEGO-body-image-1.png
alt="lego Minifigures Collection" width="300">
                     ing
src="https://i.ytimg.com/vi/nPTW-wc0aPE/maxresdefault.jpg"
alt="LEGO Duplo Train Set"
width="300"
                     These images showcase different Lego products, like bricks and figures.
                       .soc="https://pat-crawford-sdds.netlify.app/shared/winter-2025/web700/A4/sample-audio.mp3"
type="audio/mp3"
                      src="https://pat-crawford-sdds.netlify.app/shared/winter-2025/web700/A4/sample-audio.ogg" type="gudio/ogg"
                     This audio sample demonstrates how sound media is integrated on the site.
                    :|-- Video Section -->
:h3>Video</h3>
<video width="480" height="320" controls>
                       source
src="https://pat-crawford-sdds.netlify.app/shared/winter-2025/web700/A4/sample-video.mpd"
type="video/mpd"
                      src="https://pat-crawford-sdds.netlify.app/shared/winter-2025/web700/A4/sample-video.webm"
type="video/webm"
                     This video section allows us to display rich multimedia content using HTML5.
                                                                                                                             master* ↔ 🛇 0 🛕 0
```

Step 1: HTML Elements Demo (about.html)

To begin, I First wrote a short paragraph describing Myself.

Then I created a relevant header and paragraph to describe the content of the next section – AS the "Theme Demo".

Then I added Following Sections in my about.html page:

Tables Section

I added a section titled "Tables" that contains two HTML tables. Each table:

- Has at least 3 rows and 3 columns
- Uses <caption>, <thead>, , and <tfoot>
- Includes rowspan and colspan for merged cells
- Contains meaningful content such as product listings and schedules

This demonstrates how my table styles render under the new theme.

Lists Section

Next, I added a section titled "Lists" containing one ordered list () with at least 5 items. One of the items includes a nested unordered list () with 3 additional items. This showcases how list styling and spacing are handled.

Images Section

This section includes 3 images sourced from Wikimedia Commons, each with appropriate alt text and external URLs. At the bottom, I added a paragraph explaining the content of the images. These images display how the theme handles media.

Audio Section

I embedded an HTML5 audio player using the sample .mp3 and .ogg files provided in the assignment. A short paragraph explains what the audio player demonstrates.

Video Section

Lastly, I added an HTML5 video player with the provided .mp4 and .webm formats. This shows how video components are styled within the theme.

Theme.css File

```
public > css > # theme.css U X J5 server.js M J5 legoSets.js M

public > css > # theme.css > ...

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/**

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*

/*
```

```
> about.html M
                      # theme.css U X Js server.js M
                                                                 JS legoSets.js M
public > css > # theme.css > ⁴ img
        color: □#4b0082;
text-decoration: none;
      a:hover {
  color: □#2c1a5b;
        text-decoration: underline;
        margin-left: 2rem;
        padding-bottom: 1rem;
         margin-bottom: 0.5rem;
       img, video, audio {
  display: block;
        margin: 1rem auto;
        border-radius: 12px;
         box-shadow: 0 4px 12px □rgba(0, 0, 0, 0.1);
106
        background-color: ■#dedcff;
        padding: 2rem 1.5rem;
         margin-bottom: 2rem;
         border-radius: 16px;
         box-shadow: 0 6px 14px □rgba(0, 0, 0, 0.1);
naster* → ⊗ 0 🛦 0
                                                                                                                 🔍 Ln 106, Col 2 Spaces: 2 UTF-8 CRLF 🚷 CSS 🔠 🖗 Go Live
```

Step 2: CSS for custom "Theme" (theme.css)

Explanation:

I imported three different Google Fonts to give my website a custom and polished look:

- Poppins is used for general body content because it's clean and highly readable.
- Pacifico is used for headings and the navbar brand to give a friendly, handwritten feel.
- Cookie is included but not used; I initially considered it for headers but chose Pacifico instead

```
body, html {
  font-family: 'Poppins', sans-serif;
  background: linear-gradient(to bottom, #c7c6f6, #e9e9f9);
  overflow: auto;
  margin: 0;
  padding: 0;
  color: #333;
}
```

This sets the **default font** across the entire site to **Poppins** and applies a soft **purple gradient background**, as shown in the assignment's "After" screenshot. overflow: auto ensures scrollbars appear only when needed. The color sets text to dark gray for better contrast.

I customized the Bootstrap .bg-light navbar to have a soft purple gradient and a subtle shadow underneath to create depth. I used !important to override Bootstrap's default light gray background.

the navbar brand to use the Pacifico font, increasing its size and applying a soft black text shadow.

All navbar links are styled with black text and smooth hover transitions. On hover, they remain black but become underlined. I used !important to override Bootstrap's link coloring.

Headings use **Pacifico** for a consistent handwritten look. I applied a soft gray text shadow to give the text a slightly raised appearance — especially visible on the hero "About" header section.

Tables: tables follow a clean, readable style. I use alternating row colors using :nth-child(odd). Captions are bold and purple. Borders and padding create good spacing between cells.

Links: links a custom **purple** color to match the rest of the theme. When hovered, links become darker purple and are underlined.

Lists: List spacing is adjusted to make **bullets and numbers clearly readable**. The margin-left indents nested lists nicely. These styles make both ordered and unordered lists fit visually with the rest of the layout.

Media Elements (Images, Audio, Video): This styles images and media elements with:

- Center alignment
- Rounded corners
- Drop shadows

These changes make media blend smoothly into the layout.

Section: The hero section on the About page is styled with a **light purple background**, large padding, and a soft box shadow. This creates a **card-like layout** that separates content from the page background.

legoSets.js File

Step 3: Adding New Lego Sets

To extend the backend functionality, I updated my legoSets.js module (LegoData class) with a new method:

```
js
CopyEdit
addSet(newSet) {
  return new Promise((resolve, reject) => {
    const exists = this.sets.find(set => set.set_num === newSet.set_num);
    if (exists) {
      reject("Set already exists");
    } else {
      this.sets.push(newSet);
      resolve();
    }
  });
}
```

```
> about.html M
                      # theme.css U
                                           JS server.js M X JS legoSets.js M
JS server.js > ...
      const LegoData = require("./modules/legoSets");
      const legoData = new LegoData();
      const path = require('path');
      const express = require('express'); // "require" the Express module
      const app = express(); // obtain the "app" object
      const HTTP_PORT = process.env.PORT || 8081; // assign a port
      app.use(express.static(path.join(__dirname, 'public')));
      app.listen(HTTP_PORT, () => console.log(`server listening on: ${HTTP_PORT}`));
      app.get('/', (req, res) => {
       res.sendFile(path.join(__dirname, '/views/home.html'));
      app.get('/about', (req, res) => {
       res.sendFile(path.join(__dirname, '/views/about.html'));
```

```
about.html M
                       # theme.css U
                                             JS server.js M X
                                                                  JS legoSets.js M
JS server.js > ...
       app.get("/lego/add-test", async (req, res) => {
           set_num: "123",
           name: "TestSet Name",
year: "2024",
           theme_id: "366",
num_parts: "123"
           img_url: "https://fakeimg.pl/375x375?text=[+Lego+]"
          await legoData.addSet(testSet);
           res.redirect("/lego/sets");
           res.status(422).send(err);
63
       app.use((req, res) => {
        res.status(404).sendFile(path.join(__dirname, "/views/404.html"));
68
       legoData.initialize().then(() => {
         app.listen(HTTP_PORT, () => {
           console.log(`Server listening on: ${HTTP_PORT}`);
       }).catch(err => {
         console.log("Error initializing lego data:", err);
```

Step 4: Updating Your Deployment

Testing Route

To test this feature, I created the following route in server.js:

```
js
CopyEdit
app.get("/lego/add-test", async (req, res) => {
 const testSet = {
  set_num: "123",
  name: "Test Set",
  year: "2024",
  theme_id: "366",
  num parts: "123",
  img_url: "https://fakeimg.pl/375x375?text=[+Lego+]"
 };
 try {
  await legoData.addSet(testSet);
  res.redirect("/lego/sets");
 } catch (err) {
  res.status(422).send(err);
 }
});
```

When I visit /lego/add-test the first time, the new set is added and I'm redirected to the sets list. On repeat visits, the app displays an error since the set already exists — confirming the method works correctly.