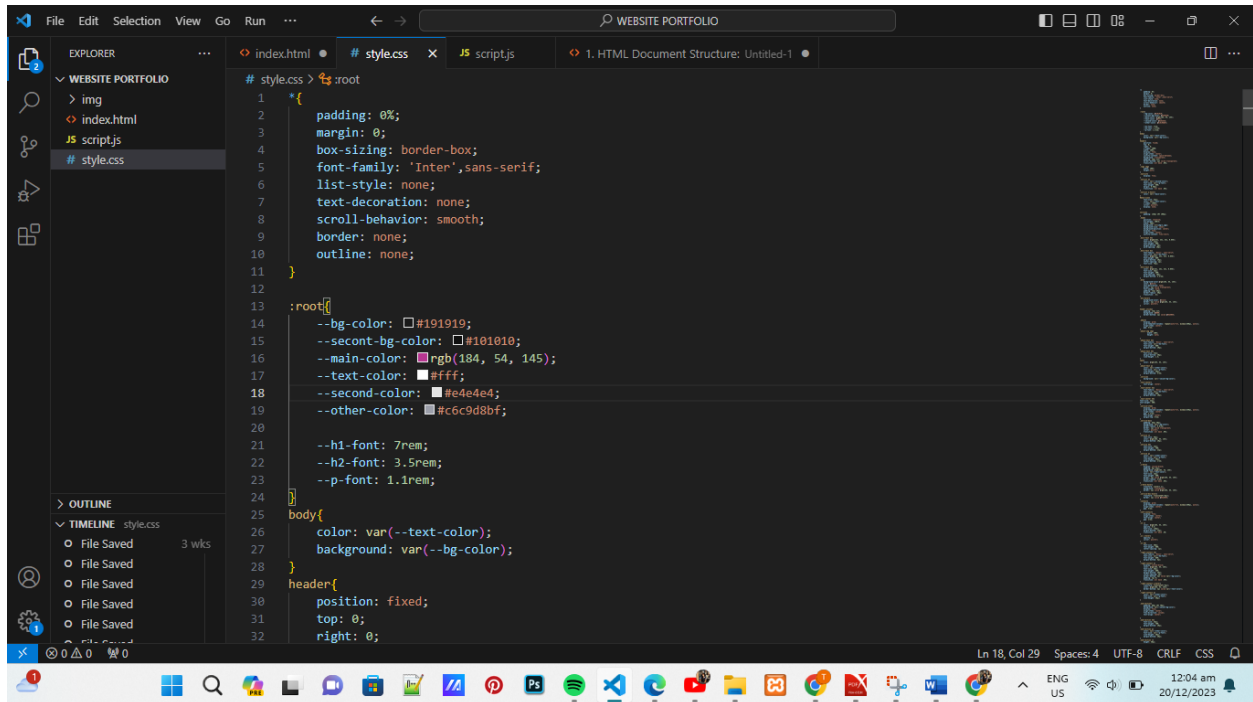


Challenge #3: Layout and Menu Elements

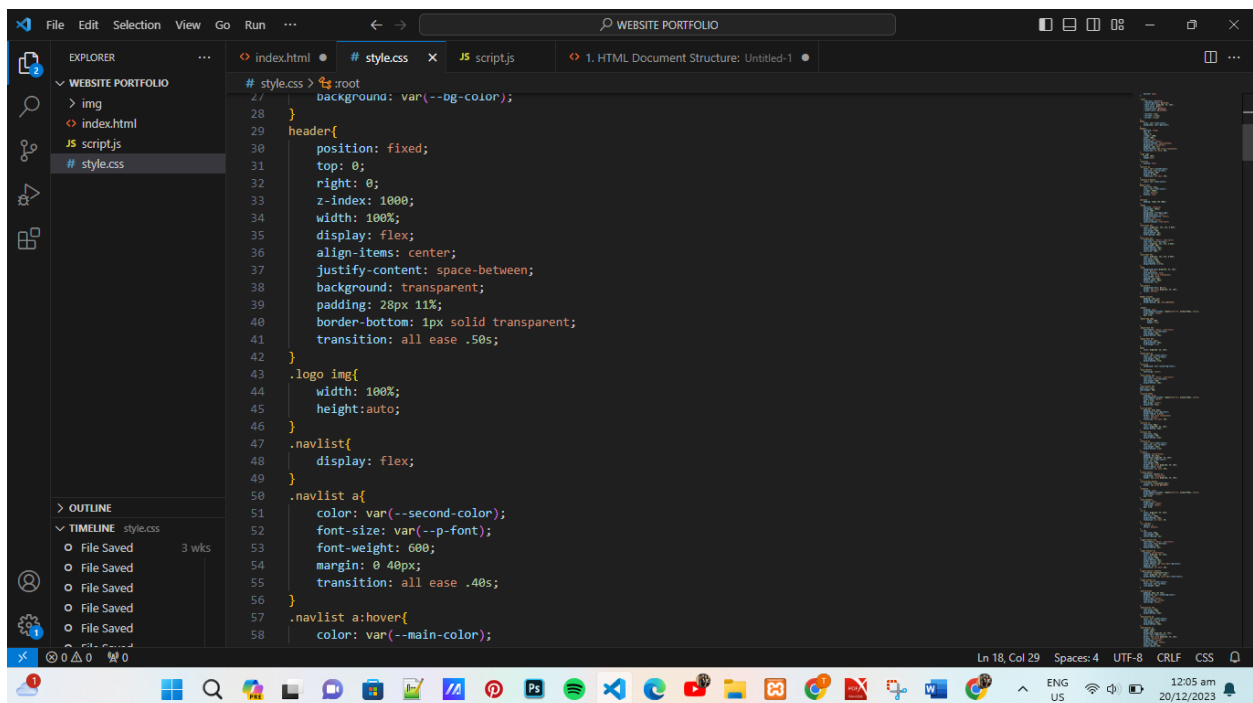
TENNESSE C. SUETA 1 BS INFOSYS-A



The screenshot shows the Visual Studio Code editor with a project named 'WEBSITE PORTFOLIO'. The Explorer sidebar on the left shows the file structure: 'img', 'index.html', 'script.js', and 'style.css'. The main editor area displays the 'style.css' file. The code defines a root selector with various reset and variable declarations, followed by a body selector and a header selector.

```
# style.css > root
1  *{
2      padding: 0%;
3      margin: 0;
4      box-sizing: border-box;
5      font-family: 'Inter', sans-serif;
6      list-style: none;
7      text-decoration: none;
8      scroll-behavior: smooth;
9      border: none;
10     outline: none;
11 }
12
13 :root{
14     --bg-color: #191919;
15     --secont-bg-color: #101010;
16     --main-color: rgb(184, 54, 145);
17     --text-color: #fff;
18     --second-color: #e4e4e4;
19     --other-color: #c6c9d8bf;
20
21     --h1-font: 7rem;
22     --h2-font: 3.5rem;
23     --p-font: 1.1rem;
24 }
25
26 body{
27     color: var(--text-color);
28     background: var(--bg-color);
29 }
30
31 header{
32     position: fixed;
33     top: 0;
34     right: 0;
```

The status bar at the bottom indicates 'Ln 18, Col 29', 'Spaces: 4', 'UTF-8', 'CRLF', 'CSS', and the system clock shows '12:04 am 20/12/2023'.

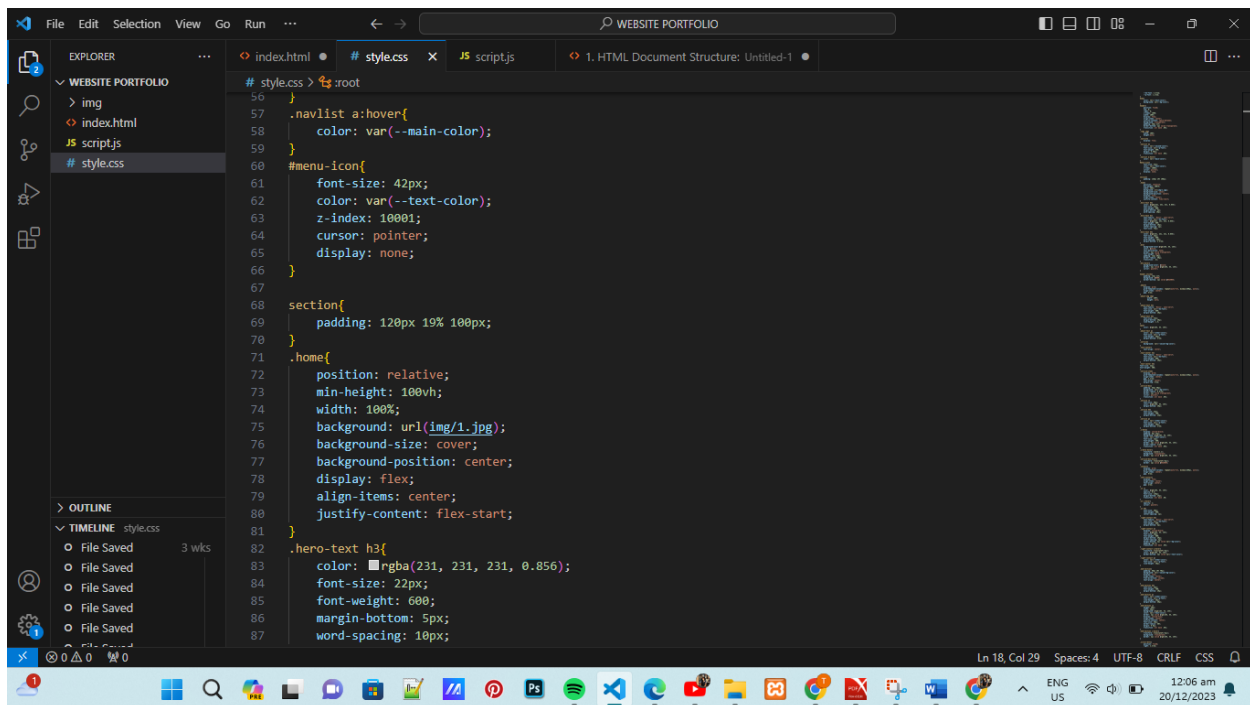


The screenshot shows the same Visual Studio Code editor with the 'style.css' file. The code continues from the previous state, adding more styles for the header, logo, and navigation list.

```

27     background: var(--bg-color);
28
29     header{
30         position: fixed;
31         top: 0;
32         right: 0;
33         z-index: 1000;
34         width: 100%;
35         display: flex;
36         align-items: center;
37         justify-content: space-between;
38         background: transparent;
39         padding: 28px 11%;
40         border-bottom: 1px solid transparent;
41         transition: all ease .50s;
42     }
43
44     .logo img{
45         width: 100%;
46         height: auto;
47     }
48
49     .navlist{
50         display: flex;
51     }
52
53     .navlist a{
54         color: var(--second-color);
55         font-size: var(--p-font);
56         font-weight: 600;
57         margin: 0 40px;
58         transition: all ease .40s;
59     }
60
61     .navlist a:hover{
62         color: var(--main-color);
63     }
64 }
```

The status bar at the bottom indicates 'Ln 18, Col 29', 'Spaces: 4', 'UTF-8', 'CRLF', 'CSS', and the system clock shows '12:05 am 20/12/2023'.



I would take the following actions to add style and menus to a planned portfolio website using what I learned from w3schools:

1. Choose a layout:

- Grid layout: Because it makes content organization simple, this design is popular for portfolio websites. It entails splitting the page into grids or parts of uniform size, with an image or project in each sector.
- Single-page layout: This style displays all of the material on one page with scrollable sections. If you wish to maintain the user's attention on your work, this is a smart option.
- Multi-page layout: This type of layout consists of several pages, each with unique content. If you want to highlight a lot of your work, this is a fantastic option.

2. Create a header:

- Logo: Position your logo in the header, ideally on the page's left side.
- Navigation menu: Design an intuitive and user-friendly navigation menu. Make sure all menu items have legible names, and that they are appropriately placed. Orient the menu to the header's right side.

3. Add a footer:

- Contact information: Include your contact information, such as your email address, phone number, and address.
- Social media links: Add links to your social media profiles, such as LinkedIn, Twitter, and Instagram.

4. Create a homepage:

- Hero section: You may use a call-to-action button that directs visitors to your portfolio page or a hero section to highlight your greatest work.
- About section: Give a brief introduction of yourself and your role. Add a captivating headline and a high-quality photo of yourself.
- Portfolio section: Use the layout (grid, single-page, or multi-page) that you have selected to display your portfolio here. For every project, use clear labeling and high-quality photos.

5. Make mobile device optimizations:

- Make use of responsive design approaches: These techniques adapt the layout of your website to the screen size of the user, ensuring that it is optimal for a wide range of screen sizes.
- Simplify the design: Make your mobile website's design simpler by eliminating extraneous components and concentrating on what's needed. Make use of legible font and sizable buttons for easier reading on small displays.