

Module -12 Assignment

Q1.) Describe the main differences between the CSS Flexbox layout model and the CSS Grid layout model. When would you choose to use one over the other?

CSS Flexbox Layout Model:-

- **One-Dimensional Layout:** Flexbox is designed for laying out items in a single direction, either as a row (horizontal) or column (vertical). It is ideal for distributing space and aligning items along one axis at a time.
- **Flexible Alignment:** Flexbox provides powerful alignment features, allowing you to easily center items vertically and horizontally, distribute space between items, or align items at the start or end of the container.
- **Responsive by Default:** Flexbox excels in responsive designs, as it adjusts its children automatically to fit the available space. Items can grow, shrink, or wrap as needed.
- **Use Case:** Best for simpler layouts where elements are aligned along one axis (horizontal or vertical). For example:
 - Navigation bars.
 - Horizontal or vertical lists of items.
 - Aligning buttons or form elements in a row or column.

CSS Grid Layout Model:-

- **Two-Dimensional Layout:** CSS Grid is designed for controlling both rows and columns simultaneously, making it ideal for more complex, grid-like layouts. You can place items across multiple rows and columns, defining a grid structure explicitly.
- **Precise Control:** With Grid, you can create precise, grid-based designs with full control over the size, spacing, and placement of items. It allows the creation of complex layouts like a dashboard or a full web page structure.
- **Explicit Layout:** You define grid containers with rows and columns, and place child items in specific grid cells, spanning multiple rows or columns if needed. This level of control allows for highly structured layouts.
- **Use Case:** Ideal for more complex, two-dimensional layouts where you need to control both rows and columns. For example:
 - Page layouts with headers, sidebars, main content, and footers.

- Image galleries or card-based grids.
- Dashboard or content-heavy layouts with sections that need exact positioning.

When to Choose Flexbox vs. Grid:-

- Use Flexbox when you:
 - Need a simple, one-direction layout (like rows or columns).
 - Want to easily align and distribute items.
 - Are working on components like navigation bars, buttons, or form layouts.
 - Require flexibility and responsive designs where elements need to wrap or adjust based on screen size.
- Use Grid when you:
 - Need a complex, two-dimensional layout.
 - Want explicit control over rows and columns.
 - Are designing entire web pages, including headers, footers, sidebars, and main content areas.
 - Need more structured and grid-based design, where positioning of items is crucial.

Q2.) Explain the role of the following key properties in the Flexbox layout model

1. justify-content

- Definition: This property aligns the flex items along the main axis (the direction in which the flex container is laid out, i.e., horizontally if flex-direction: row, or vertically if flex-direction: column).
- Role: It controls the distribution of space between and around flex items when there is extra space in the container.

Code:-

```
.container
{
  display: flex;
  justify-content: space-between;
}
```

2. align-items

- Definition: This property aligns flex items along the cross axis (perpendicular to the main axis).
- Role: It controls how items are positioned within the flex container across the cross axis (i.e., vertically for a row layout, horizontally for a column layout).

Code:-

```
.container
{
  display: flex;
  align-items: center;
}
```

3. gap

- Definition: This property sets the spacing (gap) between flex items along both the main axis and cross axis.

- Role: It is used to create space between flex items without the need for external margins on each individual item.

Values:

- The value is a length (e.g., 10px, 1em, etc.) defining the gap between items.
- You can also define row-gap and column-gap separately for additional control.

Code:-

```
.container
{
  display: flex;
  gap: 20px;
}
```

4. flex-direction

- Definition: This property defines the direction in which the flex items are placed in the flex container. It sets the main axis along which the flex items are laid out.
- Role: It controls whether items are displayed as a row (horizontally) or a column (vertically).

Code:-

```
.container
{
  display: flex;
  flex-direction: column;
}
```

5. flex-wrap

- Definition: This property controls whether flex items are forced to stay on a single line or can wrap onto multiple lines if they overflow the container.
- Role: It determines whether items should wrap to the next line when they don't fit within the container's width or height.

Code:-

```
.container
```

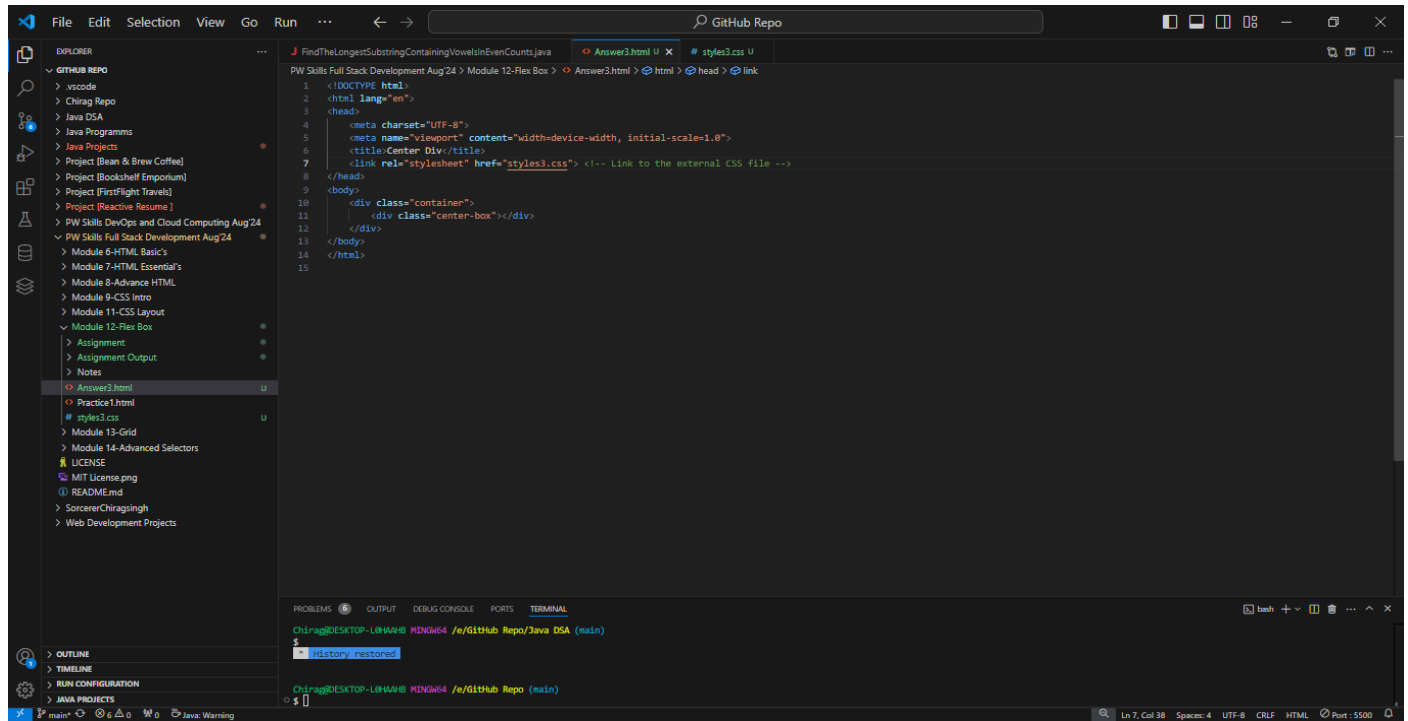
```
{
```

```
  display: flex;
```

```
  flex-wrap: wrap;
```

```
}
```

Q3.) Write the code to center a div using CSS Flexbox.

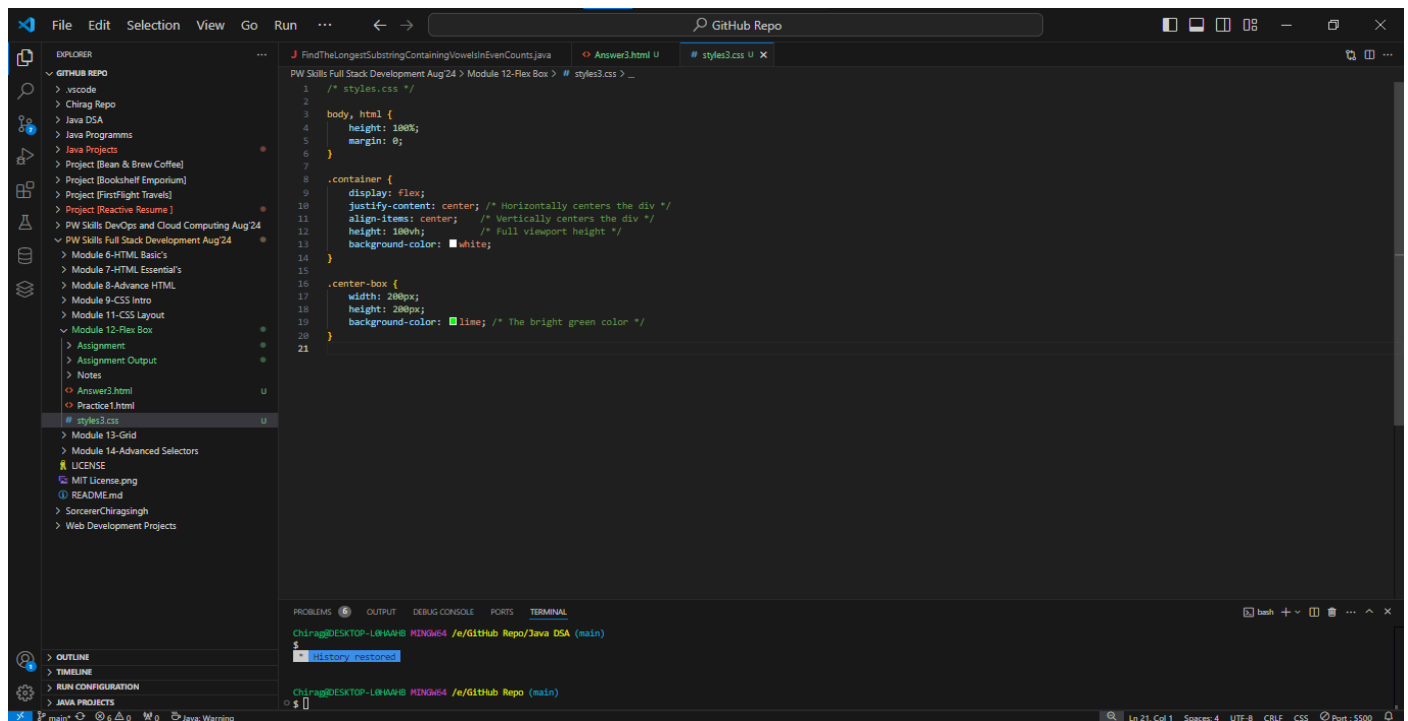


The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like 'GITLAB_REPO', 'Chirag Repo', 'Java DSA', 'Java Programs', 'Java Projects', 'Project (Bean & Brew Coffee)', 'Project (Bookshelf Emporium)', 'Project (FirstFlight Travels)', 'Project (Reactive Resume)', 'PW Skills DevOps and Cloud Computing Aug24', 'PW Skills Full Stack Development Aug24', 'Module 6-HTML Basics', 'Module 7-HTML Essential's', 'Module 8-Advance HTML', 'Module 9-CSS Intro', 'Module 11-CSS Layout', 'Module 12-Flex Box', 'Assignment', 'Assignment Output', and 'Notes'. The 'Answer3.html' file is selected in the Explorer sidebar. The main editor area shows the content of 'Answer3.html' with the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Center Div</title>
7   <link rel="stylesheet" href="styles3.css" <!-- Link to the external CSS file -->
8 </head>
9 <body>
10   <div class="container">
11     <div class="center-box"></div>
12   </div>
13 </body>
14 </html>
15
```

The bottom panel shows the TERMINAL with the following output:

```
Chirag@DESKTOP-LIHW4W: /e/GitHub Repo/Java DSA (main)
$
History restored
Chirag@DESKTOP-LIHW4W: /e/GitHub Repo (main)
$
```

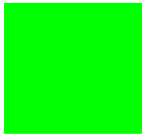


The screenshot shows the VS Code editor with the Explorer sidebar on the left. The Explorer sidebar shows a project structure with folders like 'GITLAB_REPO', 'Chirag Repo', 'Java DSA', 'Java Programs', 'Java Projects', 'Project (Bean & Brew Coffee)', 'Project (Bookshelf Emporium)', 'Project (FirstFlight Travels)', 'Project (Reactive Resume)', 'PW Skills DevOps and Cloud Computing Aug24', 'PW Skills Full Stack Development Aug24', 'Module 6-HTML Basics', 'Module 7-HTML Essential's', 'Module 8-Advance HTML', 'Module 9-CSS Intro', 'Module 11-CSS Layout', 'Module 12-Flex Box', 'Assignment', 'Assignment Output', and 'Notes'. The 'styles3.css' file is selected in the Explorer sidebar. The main editor area shows the content of 'styles3.css' with the following code:

```
1 /* styles3.css */
2
3 body, html {
4   height: 100%;
5   margin: 0;
6 }
7
8 .container {
9   display: flex;
10  justify-content: center; /* Horizontally centers the div */
11  align-items: center; /* Vertically centers the div */
12  height: 100vh; /* Full viewport height */
13  background-color: #ffffff;
14 }
15
16 .center-box {
17   width: 200px;
18   height: 200px;
19   background-color: #00ff00; /* The bright green color */
20 }
21
```

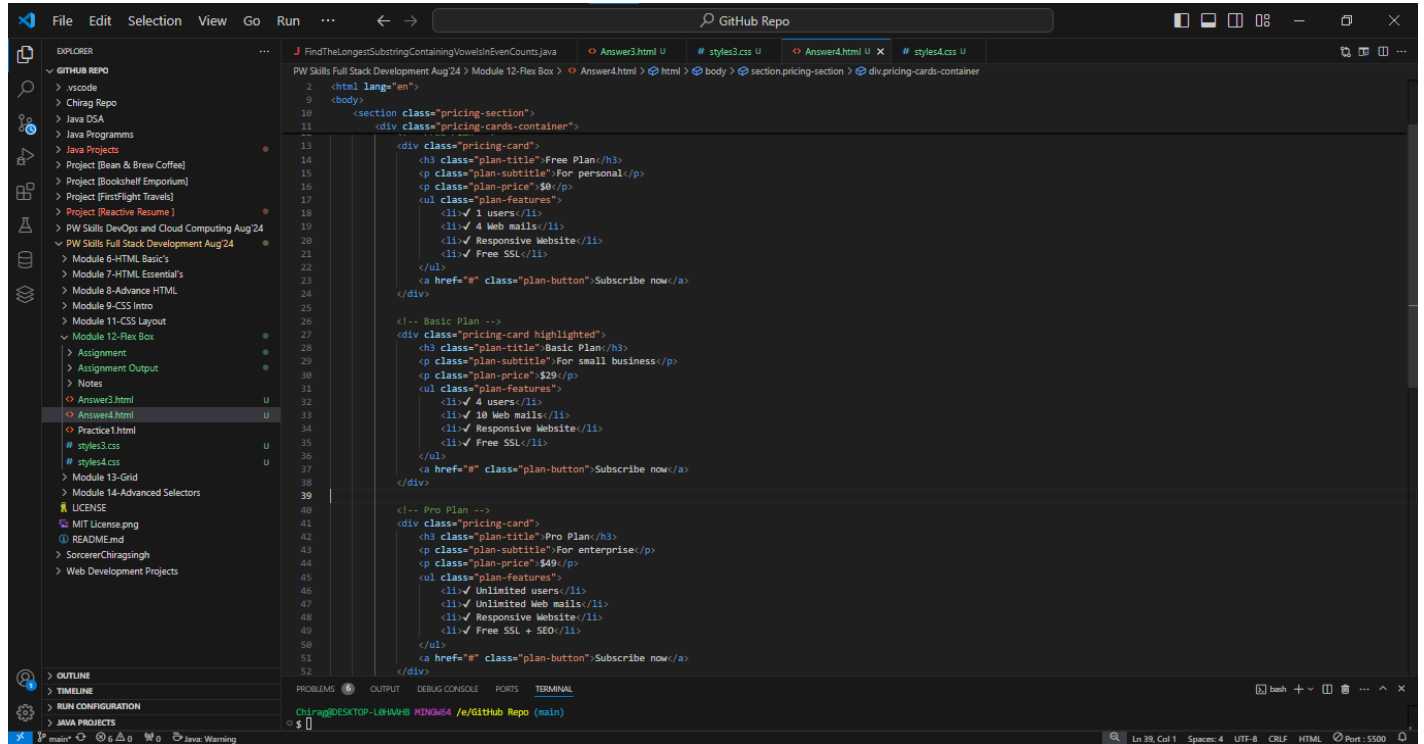
The bottom panel shows the TERMINAL with the following output:

```
Chirag@DESKTOP-LIHW4W: /e/GitHub Repo/Java DSA (main)
$
History restored
Chirag@DESKTOP-LIHW4W: /e/GitHub Repo (main)
$
```



Q4.) A client of yours wants to add a pricing section on their website to showcase their newly introduced premium plans.

You have to build the pricing section for their business. They have provided you with the figma design for the same.

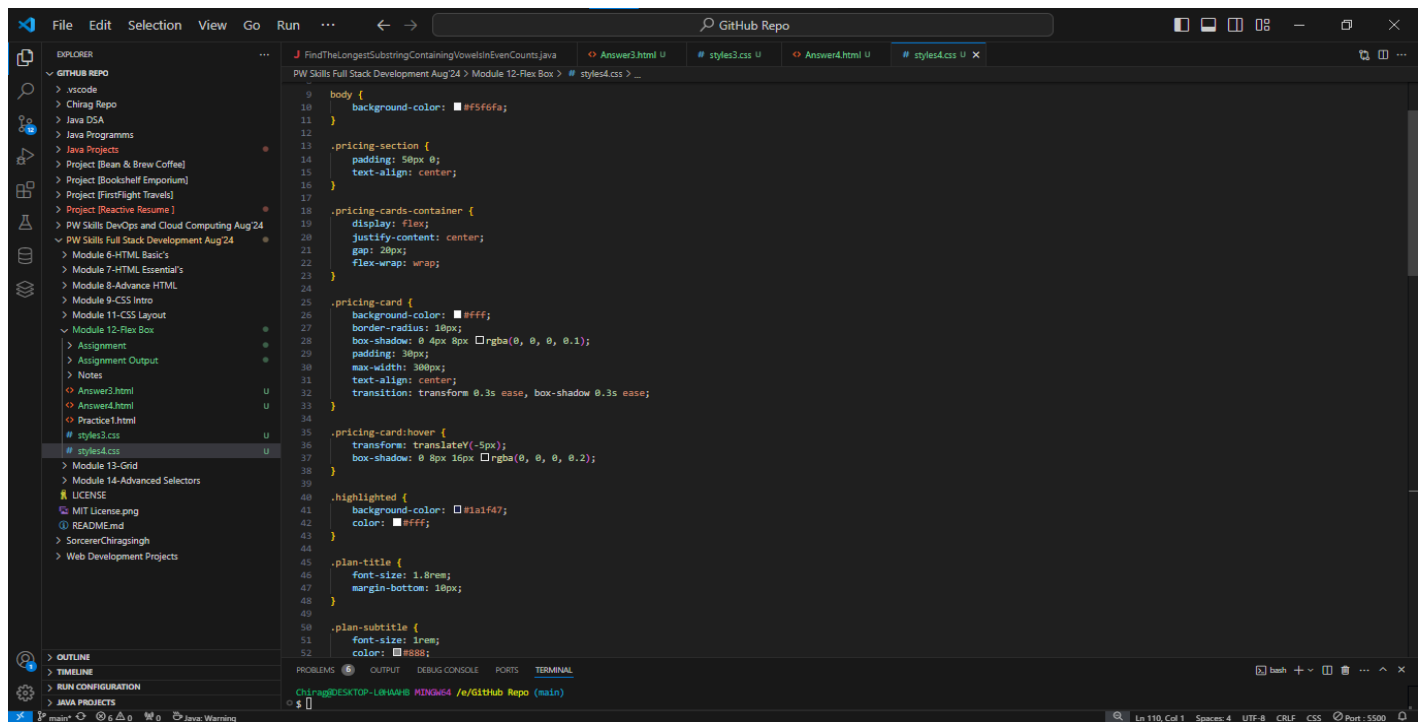


The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'chirag Repo', 'Java DSA', 'Java Programs', 'Project (Bean & Brew Coffee)', 'Project (Bookshelf Emporium)', 'Project (FirstFlight Travels)', 'Project (Reactive Resume)', 'PW Skills DevOps and Cloud Computing Aug24', 'Module 6-HTML Basic's', 'Module 7-HTML Essential's', 'Module 8-Advance HTML', 'Module 9-CSS Intro', 'Module 11-CSS Layout', 'Module 12-Flex Box', 'Assignment', 'Assignment Output', 'Notes', 'Answer2.html', 'Answer4.html', 'Practice1.html', 'style3.css', 'style4.css', 'Module 13-Grid', 'Module 14-Advanced Selectors', 'LICENSE', 'MIT License.png', 'README.md', 'SorcererChiragSingh', and 'Web Development Projects'. The code editor shows the following HTML code:

```
FindTheLongestSubstringContainingVowelsInEvenCounts.java
Answer3.html U
# style3.css U
Answer4.html U X
# style4.css U

PW Skills Full Stack Development Aug24 > Module 12-Flex Box > Answer4.html > html > body > section.pricing-section > div.pricing-cards-container
2 <html lang="en">
3 <body>
4
5 <section class="pricing-section">
6   <div class="pricing-cards-container">
7
8     <div class="pricing-card">
9       <h3 class="plan-title">Free Plan</h3>
10      <p class="plan-subtitle">For personal</p>
11      <p class="plan-price">$0</p>
12      <ul class="plan-features">
13        <li>✓ 1 users</li>
14        <li>✓ 4 Web mails</li>
15        <li>✓ Responsive Website</li>
16        <li>✓ Free SSL</li>
17      </ul>
18      <a href="#" class="plan-button">Subscribe now</a>
19    </div>
20
21    <!-- Basic Plan -->
22    <div class="pricing-card highlighted">
23      <h3 class="plan-title">Basic Plan</h3>
24      <p class="plan-subtitle">For small business</p>
25      <p class="plan-price">$29</p>
26      <ul class="plan-features">
27        <li>✓ 4 users</li>
28        <li>✓ 10 Web mails</li>
29        <li>✓ Responsive Website</li>
30        <li>✓ Free SSL</li>
31      </ul>
32      <a href="#" class="plan-button">Subscribe now</a>
33    </div>
34
35    <!-- Pro Plan -->
36    <div class="pricing-card">
37      <h3 class="plan-title">Pro Plan</h3>
38      <p class="plan-subtitle">For enterprise</p>
39      <p class="plan-price">$49</p>
40      <ul class="plan-features">
41        <li>✓ Unlimited users</li>
42        <li>✓ Unlimited Web mails</li>
43        <li>✓ Responsive Website</li>
44        <li>✓ Free SSL + SEO</li>
45      </ul>
46      <a href="#" class="plan-button">Subscribe now</a>
47    </div>
48  </div>
49
50 </section>
51
52 </body>
53 </html>
```

The terminal at the bottom shows the command 'Chirag@DESKTOP-LW6W48: MINGW64 /e/GitHub Repo (main)' and the output '\$'.



The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'chirag Repo', 'Java DSA', 'Java Programs', 'Project (Bean & Brew Coffee)', 'Project (Bookshelf Emporium)', 'Project (FirstFlight Travels)', 'Project (Reactive Resume)', 'PW Skills DevOps and Cloud Computing Aug24', 'Module 6-HTML Basic's', 'Module 7-HTML Essential's', 'Module 8-Advance HTML', 'Module 9-CSS Intro', 'Module 11-CSS Layout', 'Module 12-Flex Box', 'Assignment', 'Assignment Output', 'Notes', 'Answer2.html', 'Answer4.html', 'Practice1.html', 'style3.css', 'style4.css', 'Module 13-Grid', 'Module 14-Advanced Selectors', 'LICENSE', 'MIT License.png', 'README.md', 'SorcererChiragSingh', and 'Web Development Projects'. The code editor shows the following CSS code:

```
FindTheLongestSubstringContainingVowelsInEvenCounts.java
Answer3.html U
# style3.css U
Answer4.html U
# style4.css U X

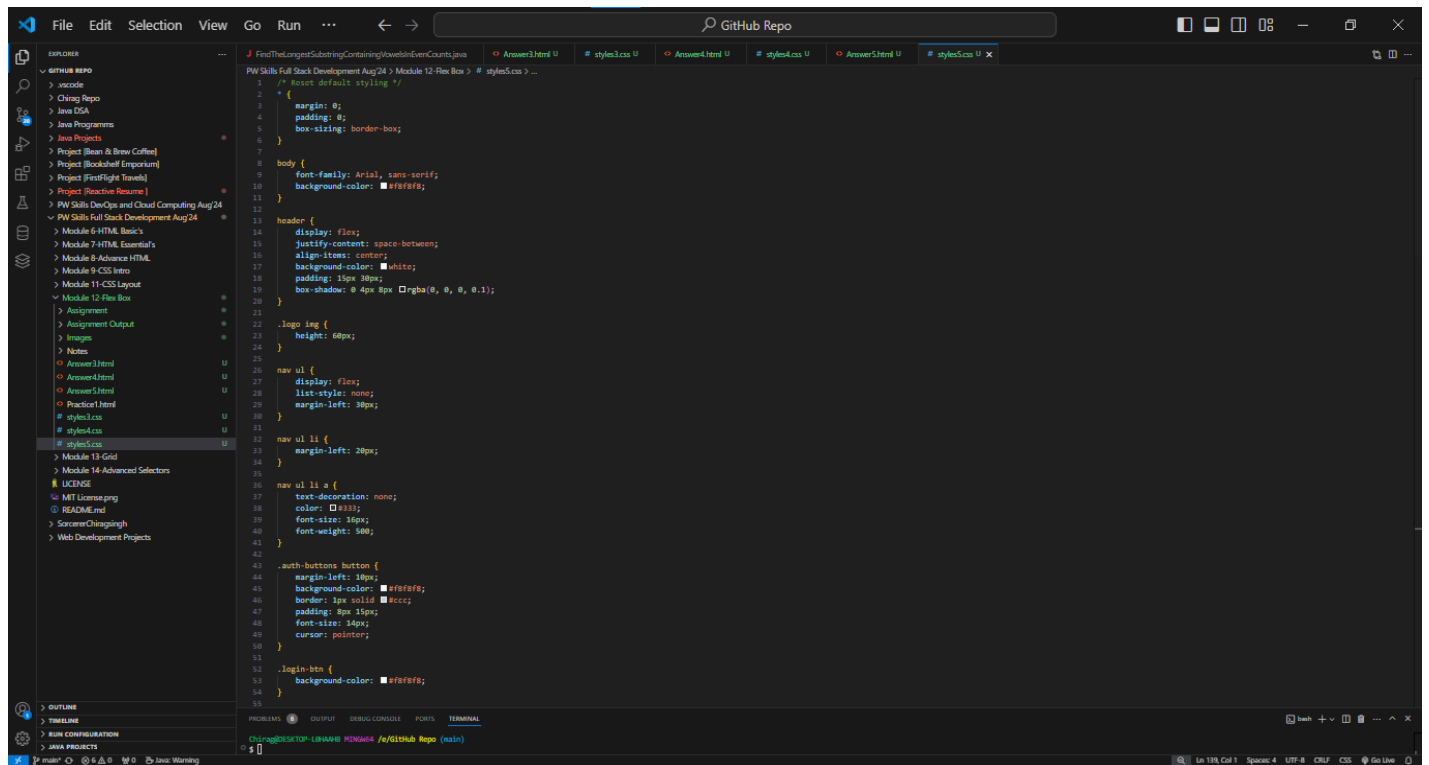
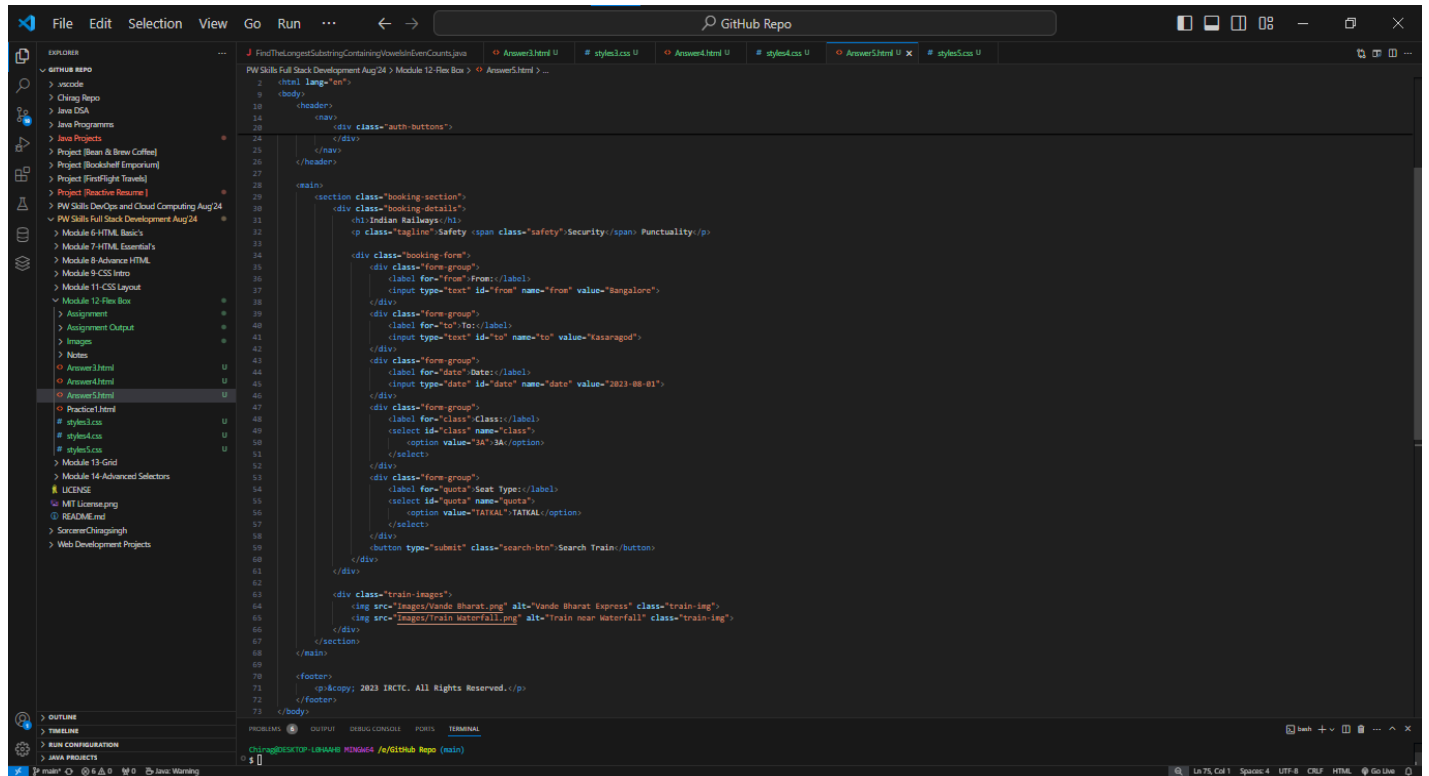
PW Skills Full Stack Development Aug24 > Module 12-Flex Box > # style4.css > ...
9 body {
10   background-color: #f5f6fa;
11 }
12
13 .pricing-section {
14   padding: 50px 0;
15   text-align: center;
16 }
17
18 .pricing-cards-container {
19   display: flex;
20   justify-content: center;
21   gap: 20px;
22   flex-wrap: wrap;
23 }
24
25 .pricing-card {
26   background-color: #fff;
27   border-radius: 10px;
28   box-shadow: 0 4px 8px #rgba(0, 0, 0, 0.1);
29   padding: 30px;
30   max-width: 300px;
31   text-align: center;
32   transition: transform 0.3s ease, box-shadow 0.3s ease;
33 }
34
35 .pricing-card:hover {
36   transform: translateY(-5px);
37   box-shadow: 0 8px 16px #rgba(0, 0, 0, 0.2);
38 }
39
40 .highlighted {
41   background-color: #e1e4f7;
42   color: #fff;
43 }
44
45 .plan-title {
46   font-size: 1.8rem;
47   margin-bottom: 10px;
48 }
49
50 .plan-subtitle {
51   font-size: 1rem;
52   color: #f888;
53 }
```

The terminal at the bottom shows the command 'Chirag@DESKTOP-LW6W48: MINGW64 /e/GitHub Repo (main)' and the output '\$'.

Free Plan For personal	Basic Plan For small business	Pro Plan For enterprise
\$0	\$29	\$49
<ul style="list-style-type: none">✓ ✓ 1 users✓ ✓ 4 Web mails✓ ✓ Responsive Website✓ ✓ Free SSL	<ul style="list-style-type: none">✓ ✓ 4 users✓ ✓ 10 Web mails✓ ✓ Responsive Website✓ ✓ Free SSL	<ul style="list-style-type: none">✓ ✓ Unlimited users✓ ✓ Unlimited Web mails✓ ✓ Responsive Website✓ ✓ Free SSL + SEO
Subscribe now	Subscribe now	Subscribe now

Q5.) Build a clone of the IRCTC Ticket booking page.

Assets can be downloaded from the figma file provided below.





Indian Railways

Safety Security Punctuality

From:

Bangalore

To:

Kasaragod

Date:

08/01/2023

Class:

3A

Seat Type:

TATKAL

Search Train

