Starting with CSS - Answer

1. Create a simple page with some div tags and show different ways to add CSS as well as what happens when you target the same elements with inline, internal, and external CSS. Also, utilize comments in the project where required.

HTML:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta  name="variable"  content="width=device-width,
initial-scale=1">
        <title>CSS Styling Methods</title>
        <link rel="stylesheet" href="style.css">
        <style>
            /*Styling for second div*/
            .internal-style{
                background-color: aqua;
                color:rgb(255, 0, 0);
                padding:10px;
                margin-bottom: 10px;
                border:1 px solid blue;
            }
            /*Style for all div elements*/
            div{
                font-family: 'Segoe UI', Tahoma, Geneva, Verdana,
sans-serif;
            }
        </style>
    </head>
    <body>
        <div  style="background-color:  brown;  color:  azure;
padding: 10px; margin-bottom: 10px;">
            This is the first div, Styled with inline CSS.
        </div>
        <div class="internal-style">
            This is the second div, Styled with internal CSS.
        </div>
        <div class="external-style">
            This is the third div, Styled with external CSS.
        </div>
        <div style="color: green;" class="internal-style external-style">
            This is fourth div, Demonstrating CSS casading.
        </div>
    </body>
</html>
```

CSS:

```
/*Style for third and fourth divs using external CSS*/
.external-style{
    background-color: yellow;
    color: green;
    padding: 20px;
    margin-bottom: 10px;
    border: 1px solid red;
    font-weight: bold;
}
/*More general style for all divs*/
div {
    border-radius: 15px;
}
```

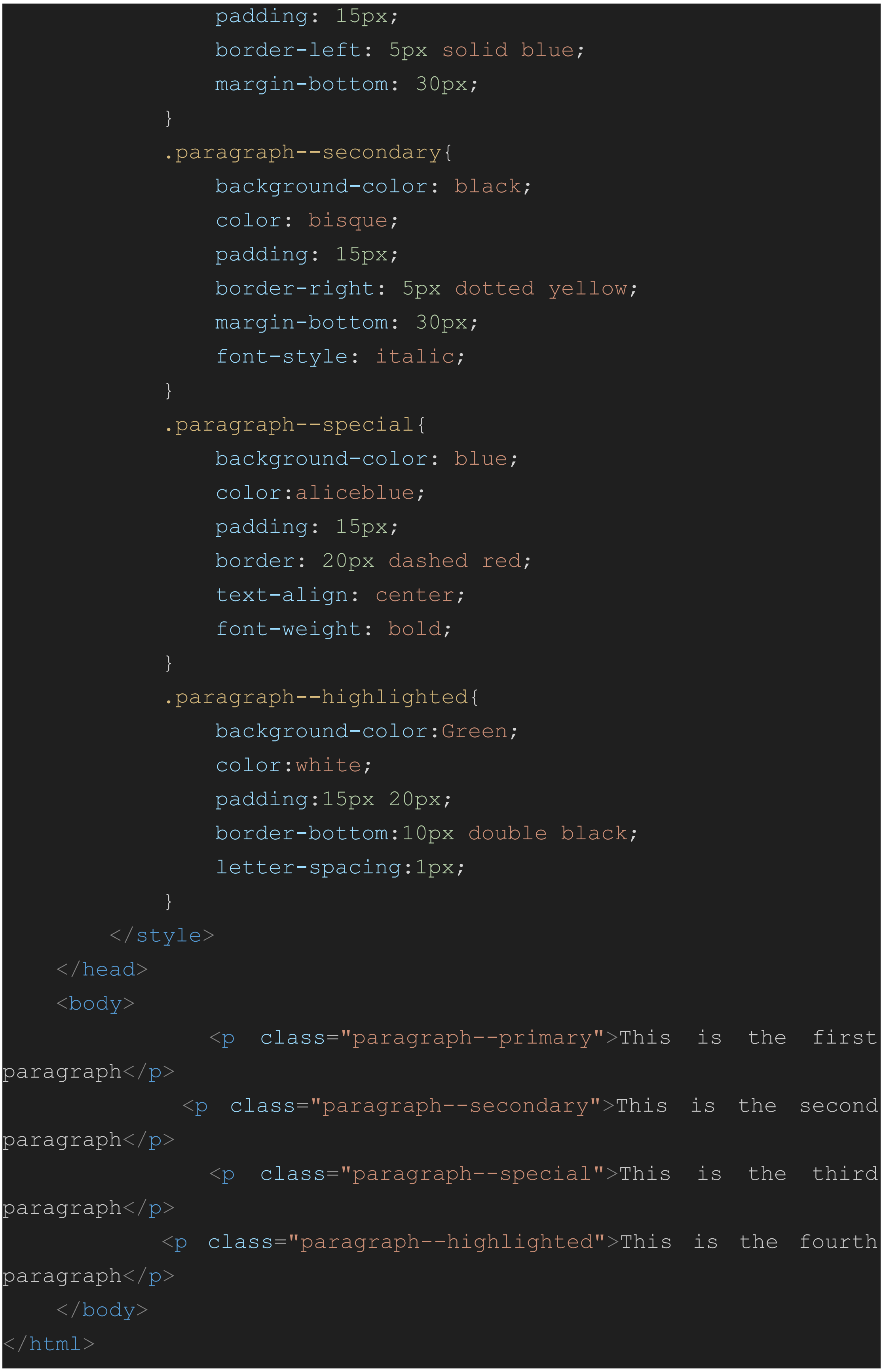
OUTPUT: Preview

---

2. Build an HTML page with multiple paragraphs, each assigned a unique class name. Write CSS rules using class selectors to apply distinct styling to each paragraph. Follow the BEM naming convention and explain how you've named the classes.

HTML & Internal-CSS:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta name="variable" content="width=device-width, initial-scale=1">
        <title>Styling Paragraphs with BEM</title>
        <style>
            .paragraph--primary{
                background-color: brown;
                color: bisque;
```



```
                padding: 15px;
                border-left: 5px solid blue;
                margin-bottom: 30px;
            }
            .paragraph--secondary{
                background-color: black;
                color: bisque;
                padding: 15px;
                border-right: 5px dotted yellow;
                margin-bottom: 30px;
                font-style: italic;
            }
            .paragraph--special{
                background-color: blue;
                color:aliceblue;
                padding: 15px;
                border: 20px dashed red;
                text-align: center;
                font-weight: bold;
            }
            .paragraph--highlighted{
                background-color:Green;
                color:white;
                padding:15px 20px;
                border-bottom:10px double black;
                letter-spacing:1px;
            }
        </style>
    </head>
    <body>
                <p class="paragraph--primary">This is the first
paragraph</p>
              <p class="paragraph--secondary">This is the second
paragraph</p>
                <p class="paragraph--special">This is the third
paragraph</p>
            <p class="paragraph--highlighted">This is the fourth
paragraph</p>
    </body>
</html>
```

Benefits of using BEM:

- **Clarity and Readability:** BEM class names are descriptive and clearly indicate the purpose of an element.
- **Specificity Management:** BEM helps to keep CSS specificity low and predictable, reducing the chances of unexpected style conflicts.
- **Reusability:** Blocks can be reused across different parts of a project, and modifiers allow for variations of those blocks.
- **Organization:** BEM provides a structured way to name CSS classes, making stylesheets easier to maintain and understand, especially in larger projects.
- **Team Collaboration:** Consistent naming conventions like BEM improve communication and collaboration among developers.

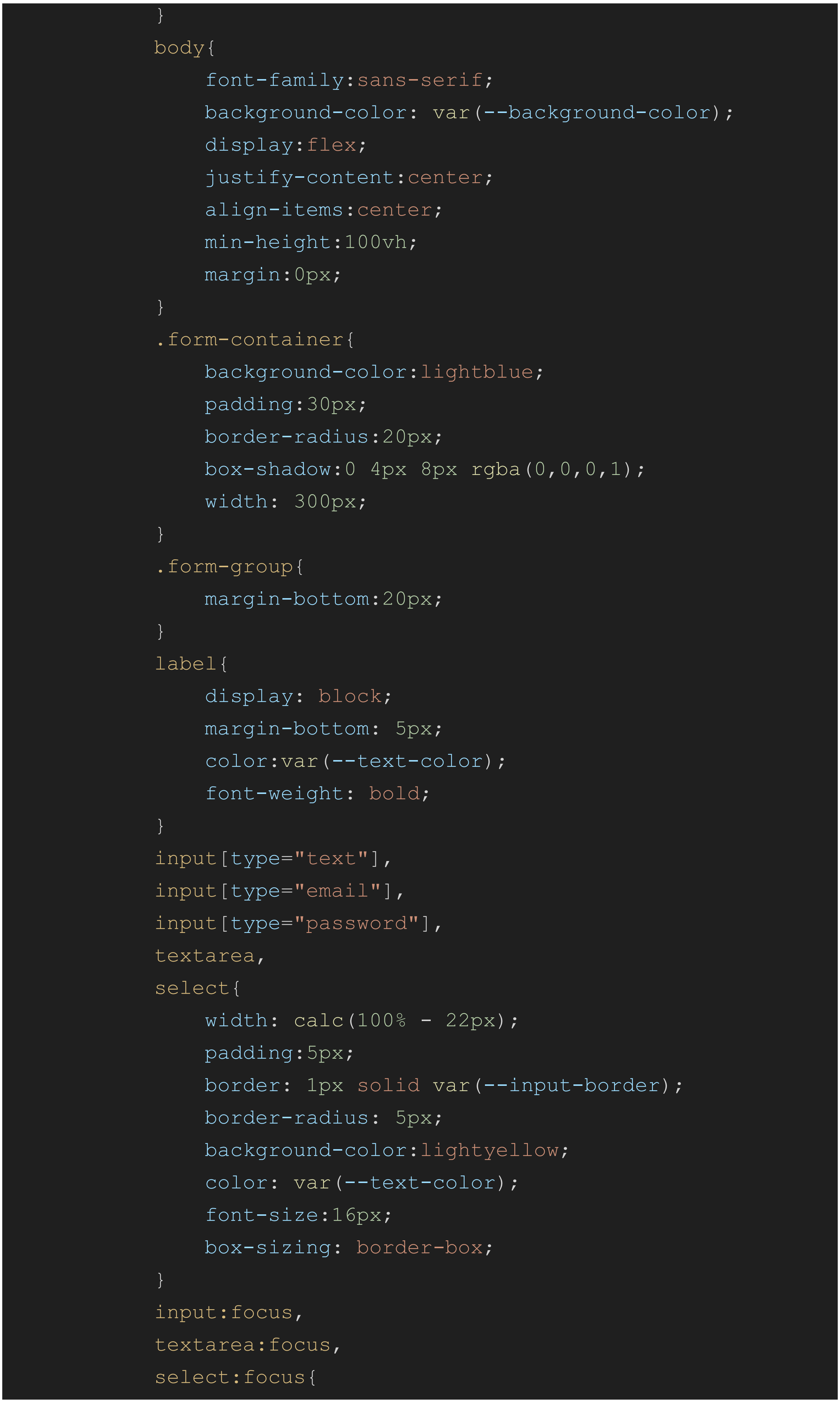
OUTPUT: Preview

---

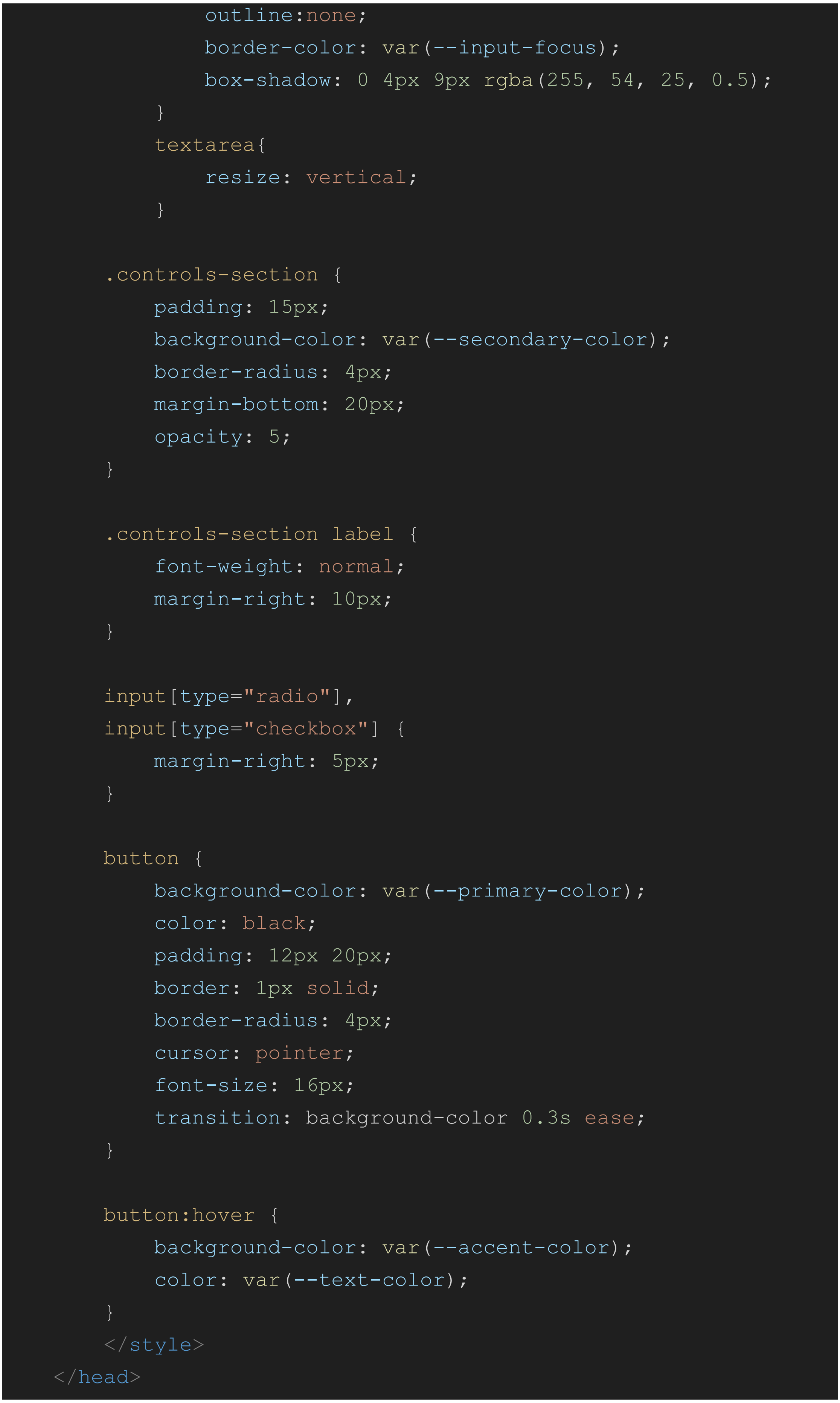
3. Develop an HTML form with various input elements. Use CSS to style the form, including setting background colors for input fields. Create a custom color palette for the form elements, and demonstrate how to apply opacity to one of the form sections.

HTML & CSS:

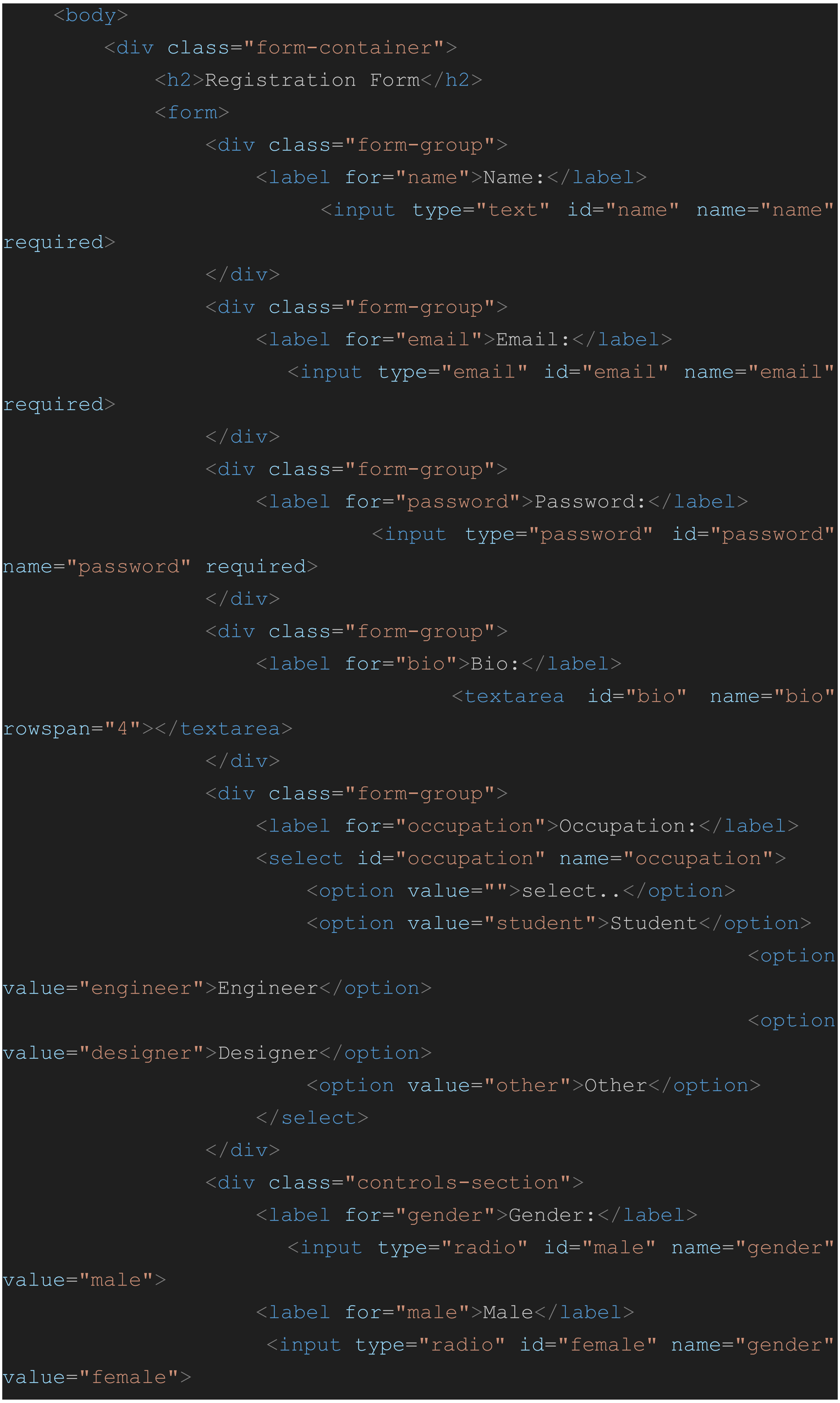
```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
              <meta  name="viewport"  content="width=device-width,
initial-scale=1">
        <title>HTML Form</title>
        <style>
            .root{
                --primary-color:#a78bfa;/*Lavender purple*/
                --secondary-color:#d8b4fe;/*Light lavender*/
                --accent-color:#ffff00;/*yellow*/
                --text-color:#708090;/*SlateGray*/
                --background-color:#f3f4f6;/*LightGray*/
                --input-background:#808080;
                --input-border:#383838;
                --input-focus:#800080;
```



```
}
body{
    font-family:sans-serif;
    background-color: var(--background-color);
    display:flex;
    justify-content:center;
    align-items:center;
    min-height:100vh;
    margin:0px;
}
.form-container{
    background-color:lightblue;
    padding:30px;
    border-radius:20px;
    box-shadow:0 4px 8px rgba(0,0,0,1);
    width: 300px;
}
.form-group{
    margin-bottom:20px;
}
label{
    display: block;
    margin-bottom: 5px;
    color:var(--text-color);
    font-weight: bold;
}
input[type="text"],
input[type="email"],
input[type="password"],
textarea,
select{
    width: calc(100% - 22px);
    padding:5px;
    border: 1px solid var(--input-border);
    border-radius: 5px;
    background-color:lightyellow;
    color: var(--text-color);
    font-size:16px;
    box-sizing: border-box;
}
input:focus,
textarea:focus,
select:focus{
```



```
            outline:none;
            border-color: var(--input-focus);
            box-shadow: 0 4px 9px rgba(255, 54, 25, 0.5);
        }
        textarea{
            resize: vertical;
        }

    .controls-section {
        padding: 15px;
        background-color: var(--secondary-color);
        border-radius: 4px;
        margin-bottom: 20px;
        opacity: 5;
    }

    .controls-section label {
        font-weight: normal;
        margin-right: 10px;
    }

    input[type="radio"],
    input[type="checkbox"] {
        margin-right: 5px;
    }

    button {
        background-color: var(--primary-color);
        color: black;
        padding: 12px 20px;
        border: 1px solid;
        border-radius: 4px;
        cursor: pointer;
        font-size: 16px;
        transition: background-color 0.3s ease;
    }

    button:hover {
        background-color: var(--accent-color);
        color: var(--text-color);
    }
    </style>
</head>
```



```
    <body>
        <div class="form-container">
            <h2>Registration Form</h2>
            <form>
                <div class="form-group">
                    <label for="name">Name:</label>
                         <input type="text" id="name" name="name"
required>
                </div>
                <div class="form-group">
                    <label for="email">Email:</label>
                      <input type="email" id="email" name="email"
required>
                </div>
                <div class="form-group">
                    <label for="password">Password:</label>
                             <input type="password" id="password"
name="password" required>
                </div>
                <div class="form-group">
                    <label for="bio">Bio:</label>
                                   <textarea id="bio" name="bio"
rowspan="4"></textarea>
                </div>
                <div class="form-group">
                    <label for="occupation">Occupation:</label>
                    <select id="occupation" name="occupation">
                        <option value="">select..</option>
                        <option value="student">Student</option>
                                                          <option
value="engineer">Engineer</option>
                                                          <option
value="designer">Designer</option>
                        <option value="other">Other</option>
                    </select>
                </div>
                <div class="controls-section">
                    <label for="gender">Gender:</label>
                      <input type="radio" id="male" name="gender"
value="male">
                    <label for="male">Male</label>
                     <input type="radio" id="female" name="gender"
value="female">
```

```
                    <label for="female">Female</label>
                            <input type="radio" id="other-gender"
name="gender" value="other">
                    <label for="other-gender">Other</label>

                    <label for="interest">Interest:</label>
                               <input type="checkbox" id="coding"
name="interest" value="coding">
                    <label for="coding">Coding</label>
                               <input type="checkbox" id="design"
name="interest" value="design">
                    <label for="design">Design</label>
                                <input type="checkbox" id="music"
name="interest" value="music">
                    <label for="music">Music</label>
                </div>
                <button type="submit">submit</button>
            </form>
        </div>
    </body>
</html>
```

Custom Color Palette (:root): We define a set of CSS variables within the :root pseudo-class. This acts as our custom color palette, making it easy to reuse and modify colors throughout the stylesheet.

- --primary-color: Used for the main action button.
- --secondary-color: Used as the background for the controls section.
- --accent-color: Used for the hover state of the button.
- --text-color: Used for the general text color.
- --background-color: Used for the body's background.
- --input-background: Used for the background of input fields.
- --input-border: Used for the border of input fields.
- --input-focus: Used for the focus state of input fields.

Basic Body Styling: We set a background color for the body and use Flexbox to center the form on the page.

Form Container Styling (.form-container): We style the container with a white background, padding, rounded corners, and a subtle box shadow to make it stand out.

Form Group Styling (.form-group): We add margin to the bottom of each form group to create vertical spacing.

Label Styling (label): We make labels display as blocks, add bottom margin, and apply the text color with bold font weight.

Input, Textarea, and Select Styling (input[type="text"], etc.):

- We set the width to fill the container (using calc to account for padding and border).
- We add padding, a border with our --input-border color, rounded corners, and the --input-background color.
- We apply the --text-color for the input text and set a base font size.
- box-sizing: border-box; ensures that padding and border are included in the element's total width and height.

Focus Styling (input:focus, etc.): When an input field, textarea, or select element is focused (clicked or tabbed into), we remove the default outline and apply a custom border color (--input-focus) and a subtle box shadow using an rgba value to include opacity.

Textarea Resizing: We allow vertical resizing of the textarea.

Controls Section Styling (.controls-section):

- We add padding and set the background color to our --secondary-color.
- We apply border-radius for rounded corners.
- We set opacity: 0.8; to make this entire section slightly transparent, allowing the background to show through a bit.

Radio and Checkbox Styling (input[type="radio"], input[type="checkbox"]): We add some right margin for spacing between the input and the label.

Button Styling (button):

- We set the background color to our --primary-color and the text color to white.
- We add padding, remove the border, apply rounded corners, and set the cursor to pointer.
- We add a transition property for a smooth effect on hover.

Button Hover Styling (button:hover): When the mouse hovers over the button, we change its background color to our --accent-color and the text color to the --text-color