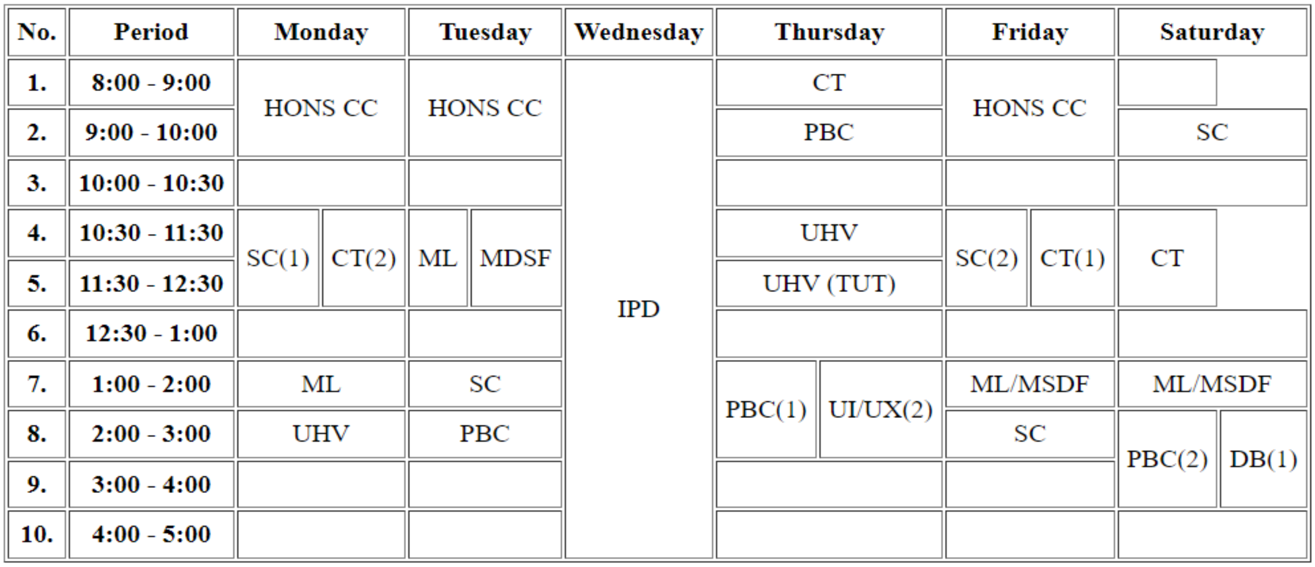
**Name: Ashish Maurya** **SAP ID: 60019220030**

**Branch: CSE (ICB)** **Div: B1**

**Course: Web Application Development Laboratory**

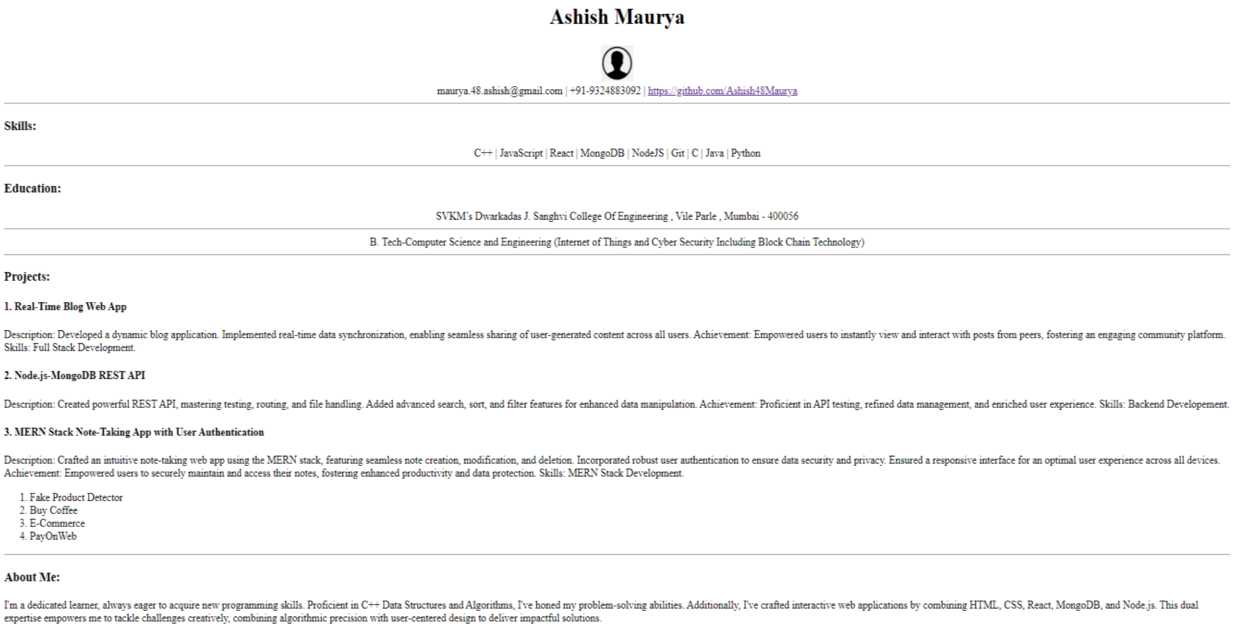
# EXPERIMENT NO. 1

A) Create a Class Timetable using HTML.



<table>
<tr><th>No.</th><th>Period</th><th colspan="2">Monday</th><th colspan="2">Tuesday</th><th>Wednesday</th><th colspan="2">Thursday</th><th colspan="2">Friday</th><th colspan="2">Saturday</th></tr>
<tr><td>1.</td><td>8:00 - 9:00</td><td colspan="2" rowspan="2">HONS CC</td><td colspan="2" rowspan="2">HONS CC</td><td rowspan="10">IPD</td><td colspan="2">CT</td><td colspan="2" rowspan="2">HONS CC</td><td></td><td></td></tr>
<tr><td>2.</td><td>9:00 - 10:00</td><td colspan="2">PBC</td><td colspan="2">SC</td></tr>
<tr><td>3.</td><td>10:00 - 10:30</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>4.</td><td>10:30 - 11:30</td><td rowspan="2">SC(1)</td><td rowspan="2">CT(2)</td><td rowspan="2">ML</td><td rowspan="2">MDSF</td><td colspan="2">UHV</td><td rowspan="2">SC(2)</td><td rowspan="2">CT(1)</td><td rowspan="2">CT</td><td rowspan="2"></td></tr>
<tr><td>5.</td><td>11:30 - 12:30</td><td colspan="2">UHV (TUT)</td></tr>
<tr><td>6.</td><td>12:30 - 1:00</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>7.</td><td>1:00 - 2:00</td><td colspan="2">ML</td><td colspan="2">SC</td><td rowspan="2">PBC(1)</td><td rowspan="2">UI/UX(2)</td><td colspan="2">ML/MSDF</td><td colspan="2">ML/MSDF</td></tr>
<tr><td>8.</td><td>2:00 - 3:00</td><td colspan="2">UHV</td><td colspan="2">PBC</td><td colspan="2">SC</td><td rowspan="2">PBC(2)</td><td rowspan="2">DB(1)</td></tr>
<tr><td>9.</td><td>3:00 - 4:00</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
<tr><td>10.</td><td>4:00 - 5:00</td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td><td colspan="2"></td></tr>
</table>

B) Create a Resume HTML.



## Ashish Maurya

maurya.48.ashish@gmail.com | +91-9324883092 | https://github.com/Ashish48Maurya

**Skills:**

C++ | JavaScript | React | MongoDB | NodeJS | Git | C | Java | Python

**Education:**

SVKM's Dwarkadas J. Sanghvi College Of Engineering , Vile Parle , Mumbai - 400056

B. Tech-Computer Science and Engineering (Internet of Things and Cyber Security Including Block Chain Technology)

**Projects:**

**1. Real-Time Blog Web App**

Description: Developed a dynamic blog application. Implemented real-time data synchronization, enabling seamless sharing of user-generated content across all users. Achievement: Empowered users to instantly view and interact with posts from peers, fostering an engaging community platform. Skills: Full Stack Development.

**2. Node.js-MongoDB REST API**

Description: Created powerful REST API, mastering testing, routing, and file handling. Added advanced search, sort, and filter features for enhanced data manipulation. Achievement: Proficient in API testing, refined data management, and enriched user experience. Skills: Backend Developement.

**3. MERN Stack Note-Taking App with User Authentication**

Description: Crafted an intuitive note-taking web app using the MERN stack, featuring seamless note creation, modification, and deletion. Incorporated robust user authentication to ensure data security and privacy. Ensured a responsive interface for an optimal user experience across all devices. Achievement: Empowered users to securely maintain and access their notes, fostering enhanced productivity and data protection. Skills: MERN Stack Development.

1. Fake Product Detector
2. Buy Coffee
3. E-Commerce
4. PayOnWeb

**About Me:**

I'm a dedicated learner, always eager to acquire new programming skills. Proficient in C++ Data Structures and Algorithms, I've honed my problem-solving abilities. Additionally, I've crafted interactive web applications by combining HTML, CSS, React, MongoDB, and Node.js. This dual expertise empowers me to tackle challenges creatively, combining algorithmic precision with user-centered design to deliver impactful solutions.

Theory:

1) Explain the different HTML elements used to implement class timetable, Resume web page using HTML tags.

| | |
|---|---|
| Defines the document type | <!DOCTYPE> |
| Defines an HTML document | <html> |
| Contains metadata/information for the document | <head> |
| Defines a title for the document | <title> |
| Defines the document's body | <body> |
| Defines HTML headings | <h1> to <h6> |
| Defines a paragraph | <p> |
| Inserts a single line break | <br> |
| Defines a thematic change in the content | <hr> |
| Defines a comment | <!--...--> |
| Defines an image | <img> |
| Defines a hyperlink | <a> |
| Defines the relationship b/w doc and ext. resource | <link> |
| Defines a list item | <li> |
| Defines a table | <table> |
| Defines a table caption | <caption> |
| Defines a header cell in a table | <th> |
| Defines a row in a table | <tr> |
| Defines a cell in a table | <td> |
| Defines style information for a document | <style> |
| Defines a section in a document | <div> |
| Defines a section in a document | <span> |
| Defines a header for a document or section | <header> |
| Defines a header and related content | <hgroup> |
| Defines a footer for a document or section | <footer> |
| Specifies the main content of a document | <main> |
| Defines metadata about an HTML document | <meta> |

2) Differentiate HTML and HTML5.

| HTML | HTML5 |
|---|---|
| It didn't support audio and video without the use of flash player support. | It supports audio and video controls with the use of <audio> and <video> tags. |
| It uses cookies to store temporary data. | It uses SQL databases and application cache to store offline data. |
| It does not allow drag and drop effects. | It allows drag and drop effects. |
| It works with all old browsers. | It works with all new browsers. |
| Doctype declaration is too long and complicated. | Doctype declaration is quite simple and easy. |
| Character encoding is long and complicated. | Character encoding is small and easy. |
| It can not handle inaccurate syntax. | It can handle inaccurate syntax. |
| Does not allow JavaScript to run in the browser. | Does allow JavaScript to run in the background. |
| Attributes like charset, async and ping are absent in HTML. | Attributes like charset, async and ping are present in HTML5. |

**New HTML5 Tags:**

<article> Represents an independent piece of content of a document, such as a blog entry or newspaper article

<aside > Represents a piece of content that is only slightly related to the rest of the page.

<audio> Defines an audio file.

<canvas> This is used for rendering dynamic bitmap graphics on the fly, such as graphs or games.

<command> Represents a command the user can invoke.

Conclusion: Thus we successfully learnt various HTML Tags and Elements & understood the difference between HTML & HTML 5.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Name: Parshav Dedhia** **SAP ID: 60019220120**

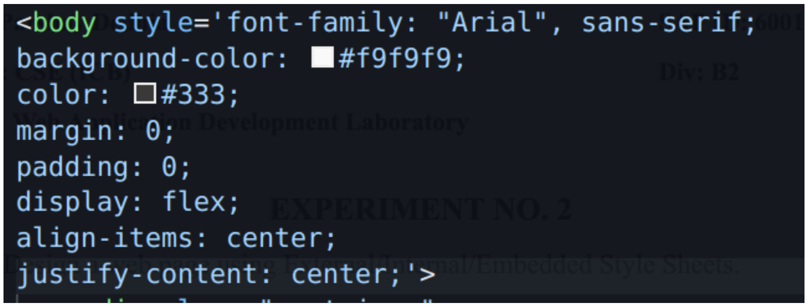
**Branch: CSE (ICB)** **Div: B2**

**Course: Web Application Development Laboratory**

# EXPERIMENT NO. 2

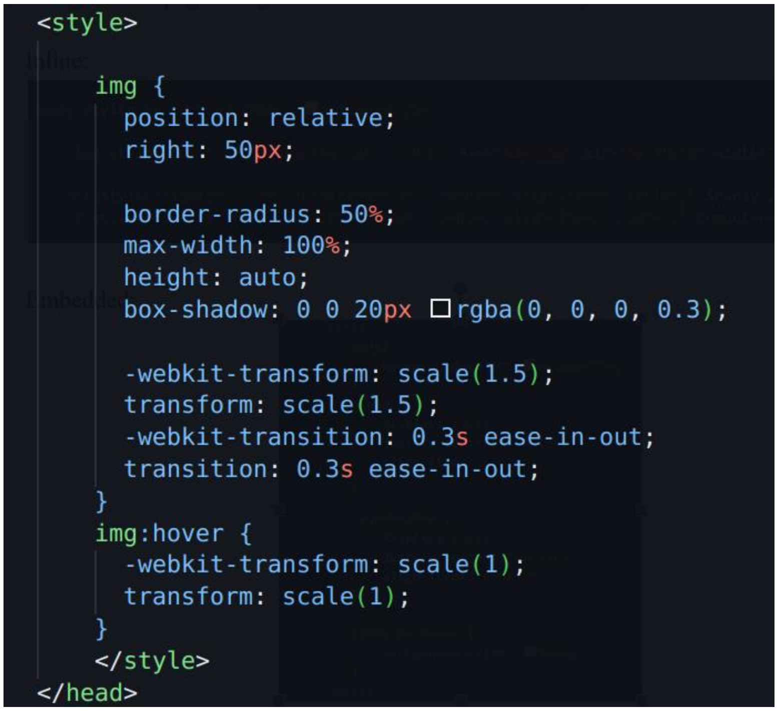
A) Design a web page using External/Internal/Embedded Style Sheets.

Inline:



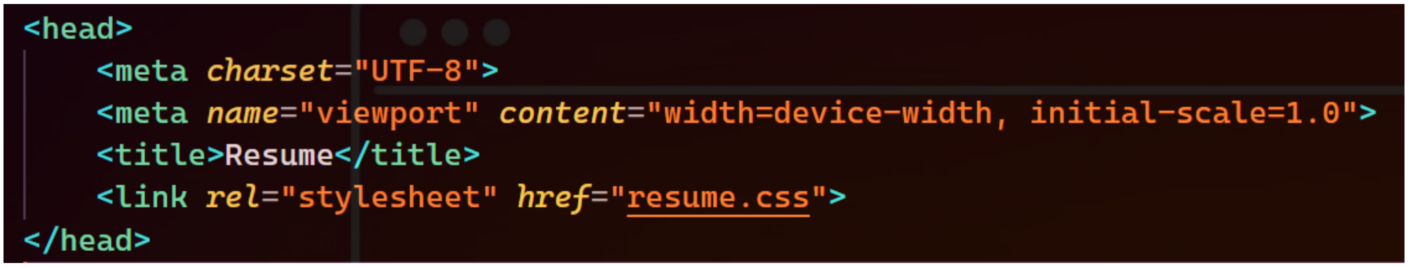
```
<body style='font-family: "Arial", sans-serif;
background-color: □#f9f9f9;
color: □#333;
margin: 0;
padding: 0;
display: flex;
align-items: center;
justify-content: center;'>
```

Embedded:

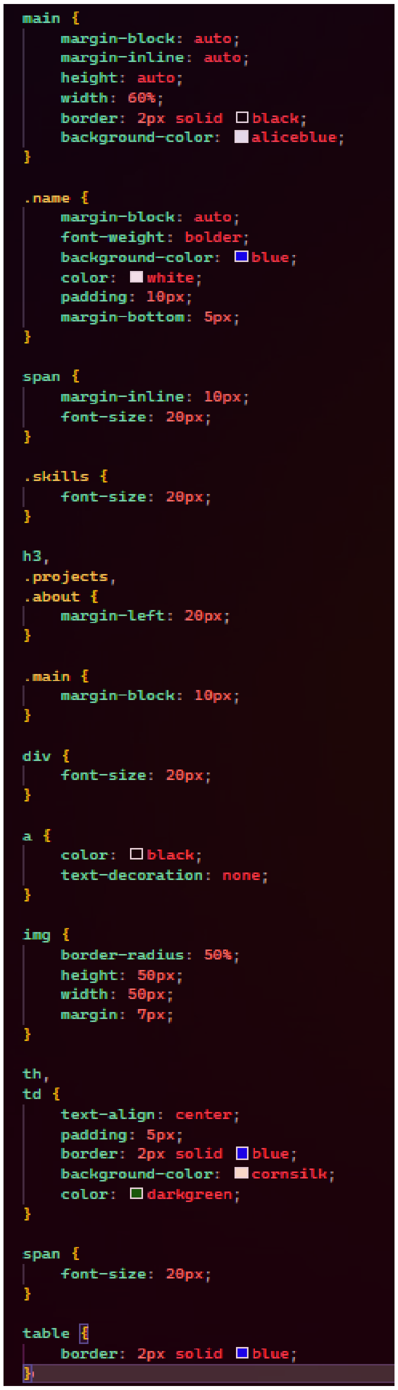


```
<style>

    img {
      position: relative;
      right: 50px;

      border-radius: 50%;
      max-width: 100%;
      height: auto;
      box-shadow: 0 0 20px □rgba(0, 0, 0, 0.3);

      -webkit-transform: scale(1.5);
      transform: scale(1.5);
      -webkit-transition: 0.3s ease-in-out;
      transition: 0.3s ease-in-out;
    }
    img:hover {
      -webkit-transform: scale(1);
      transform: scale(1);
    }
    </style>
</head>
```

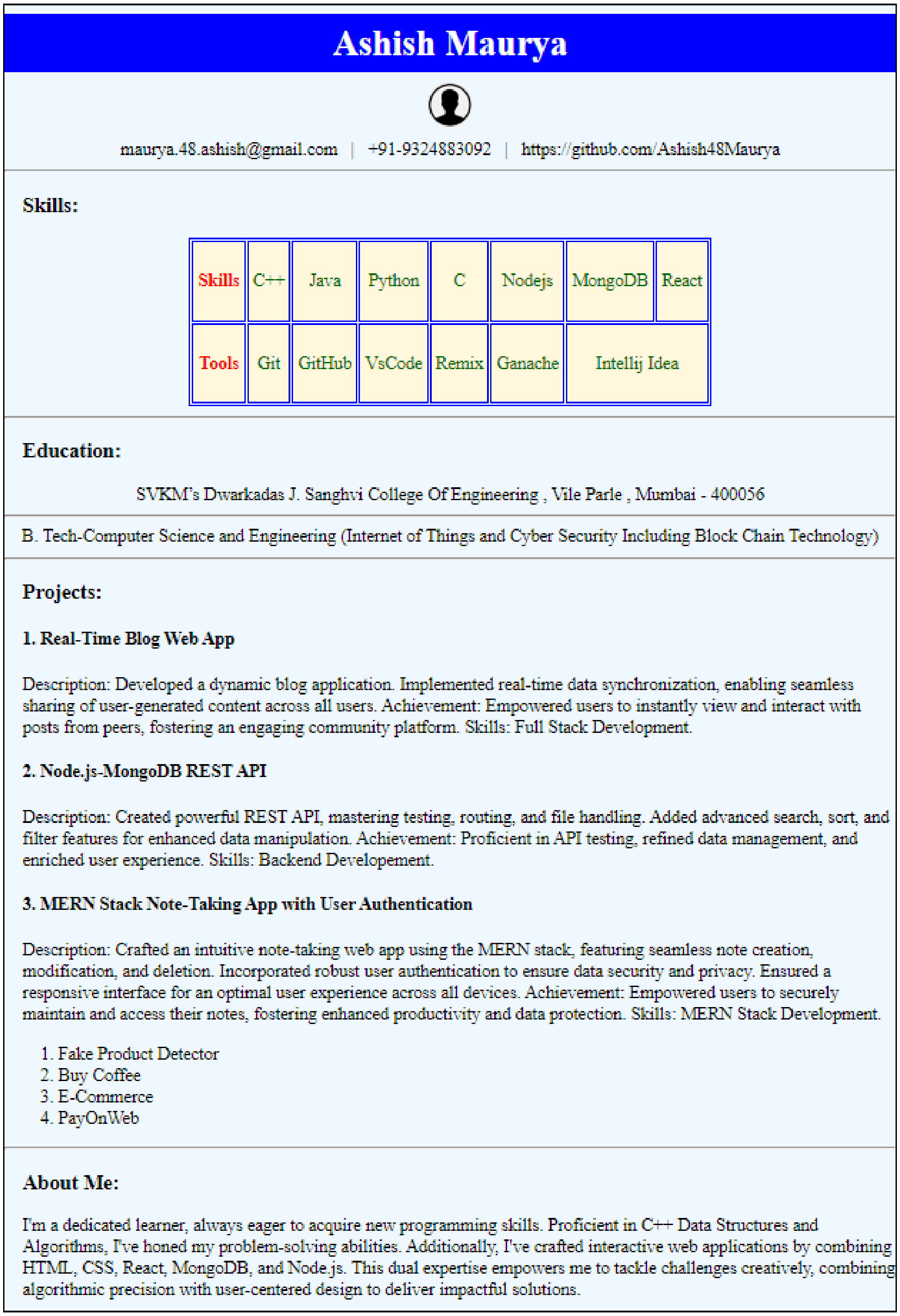
External:



```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Resume</title>
    <link rel="stylesheet" href="resume.css">
</head>
```



```
main {
    margin-block: auto;
    margin-inline: auto;
    height: auto;
    width: 60%;
    border: 2px solid black;
    background-color: aliceblue;
}

.name {
    margin-block: auto;
    font-weight: bolder;
    background-color: blue;
    color: white;
    padding: 10px;
    margin-bottom: 5px;
}

span {
    margin-inline: 10px;
    font-size: 20px;
}

.skills {
    font-size: 20px;
}

h3,
.projects,
.about {
    margin-left: 20px;
}

.main {
    margin-block: 10px;
}

div {
    font-size: 20px;
}

a {
    color: black;
    text-decoration: none;
}

img {
    border-radius: 50%;
    height: 50px;
    width: 50px;
    margin: 7px;
}

th,
td {
    text-align: center;
    padding: 5px;
    border: 2px solid blue;
    background-color: cornsilk;
    color: darkgreen;
}

span {
    font-size: 20px;
}

table {
    border: 2px solid blue;
}
```



# Ashish Maurya

maurya.48.ashish@gmail.com | +91-9324883092 | https://github.com/Ashish48Maurya

## Skills:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Skills | C++ | Java | Python | C | Nodejs | MongoDB | React |
| Tools | Git | GitHub | VsCode | Remix | Ganache | Intellij Idea | |

## Education:

SVKM's Dwarkadas J. Sanghvi College Of Engineering , Vile Parle , Mumbai - 400056

B. Tech-Computer Science and Engineering (Internet of Things and Cyber Security Including Block Chain Technology)

## Projects:

**1. Real-Time Blog Web App**

Description: Developed a dynamic blog application. Implemented real-time data synchronization, enabling seamless sharing of user-generated content across all users. Achievement: Empowered users to instantly view and interact with posts from peers, fostering an engaging community platform. Skills: Full Stack Development.

**2. Node.js-MongoDB REST API**

Description: Created powerful REST API, mastering testing, routing, and file handling. Added advanced search, sort, and filter features for enhanced data manipulation. Achievement: Proficient in API testing, refined data management, and enriched user experience. Skills: Backend Developement.

**3. MERN Stack Note-Taking App with User Authentication**

Description: Crafted an intuitive note-taking web app using the MERN stack, featuring seamless note creation, modification, and deletion. Incorporated robust user authentication to ensure data security and privacy. Ensured a responsive interface for an optimal user experience across all devices. Achievement: Empowered users to securely maintain and access their notes, fostering enhanced productivity and data protection. Skills: MERN Stack Development.

1. Fake Product Detector
2. Buy Coffee
3. E-Commerce
4. PayOnWeb

## About Me:

I'm a dedicated learner, always eager to acquire new programming skills. Proficient in C++ Data Structures and Algorithms, I've honed my problem-solving abilities. Additionally, I've crafted interactive web applications by combining HTML, CSS, React, MongoDB, and Node.js. This dual expertise empowers me to tackle challenges creatively, combining algorithmic precision with user-centered design to deliver impactful solutions.

Theory:

1) Explain different tags used to include Inline/Embedded /External Style Sheets in HTML.

Inline Styles:

Inline styles are applied directly to individual HTML elements using the <style> attribute. The <style> attribute contains CSS declarations for the specific element.

Example:

```
<p style="color: blue; font-size: 16px;">
This is a paragraph with inline styles. </p>
```

Embedded Styles:

Embedded styles are defined within the HTML document, typically within the <head> section, using the <style> tag. The <style> tag includes the CSS rules for styling the document.

Example:

```
<!DOCTYPE html>
<html>
<head>
   <style>
      p {
         color: red;
         font-size: 18px;
      }
   </style>
</head>
<body>
   <p>This is a paragraph with embedded styles.</p>
</body>
</html>
```

External Styles:

External styles are stored in separate CSS files and linked to the HTML document using the <link> tag. The <link> tag is placed within the <head> section of the HTML document. The href attribute in the <link> tag specifies the path to the external CSS file.

Example:

```
<!DOCTYPE html>
<html>
<head>
   <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
   <p>This is a paragraph with external styles.</p>
</body>
</html>
```

In the external styles example, the styles.css file would contain the CSS rules, such as:

```
/* styles.css */
p {
   color: green;
   font-size: 20px;
}
```

2) Explain Cascading Order.

The term "cascading" in Cascading Style Sheets (CSS) refers to the order of priority or precedence that the browser follows when applying styles to an HTML document. The cascading order is defined by the CSS specification and consists of different levels, each with its own weight. The order of importance, from highest to lowest, is often remembered using the acronym "CASCADING." Inline styles have the highest priority and will override any styles declared elsewhere. Next are internal styles, which follow inline styles. External stylesheets come last and will be overridden by inline and internal styles.

3) Types of CSS Selectors.
CSS (Cascading Style Sheets) selectors are patterns used to select and style HTML elements. There are various types of CSS selectors, each with its own specificity and use cases. Here are some common types of CSS selectors:

   a) Universal Selector (*)
   Selects all elements on the page.
   Example: * { margin: 0; padding: 0;
   }

   b) Type or Element Selector
   Selects all instances of a specified HTML element.
   Example: p { color: blue; }

   c) Class Selector (.)
   Selects elements with a specific class attribute.
   Example: .highlight { background-color: yellow; }

   d) ID Selector (#)
   Selects a single element with a specific ID attribute.
   Example: #header { font-size: 24px; }

   e) Descendant Selector (space)
   Selects all descendants of a specified element, regardless of their depth in the document tree.
   Example: article p { font-style: italic; }

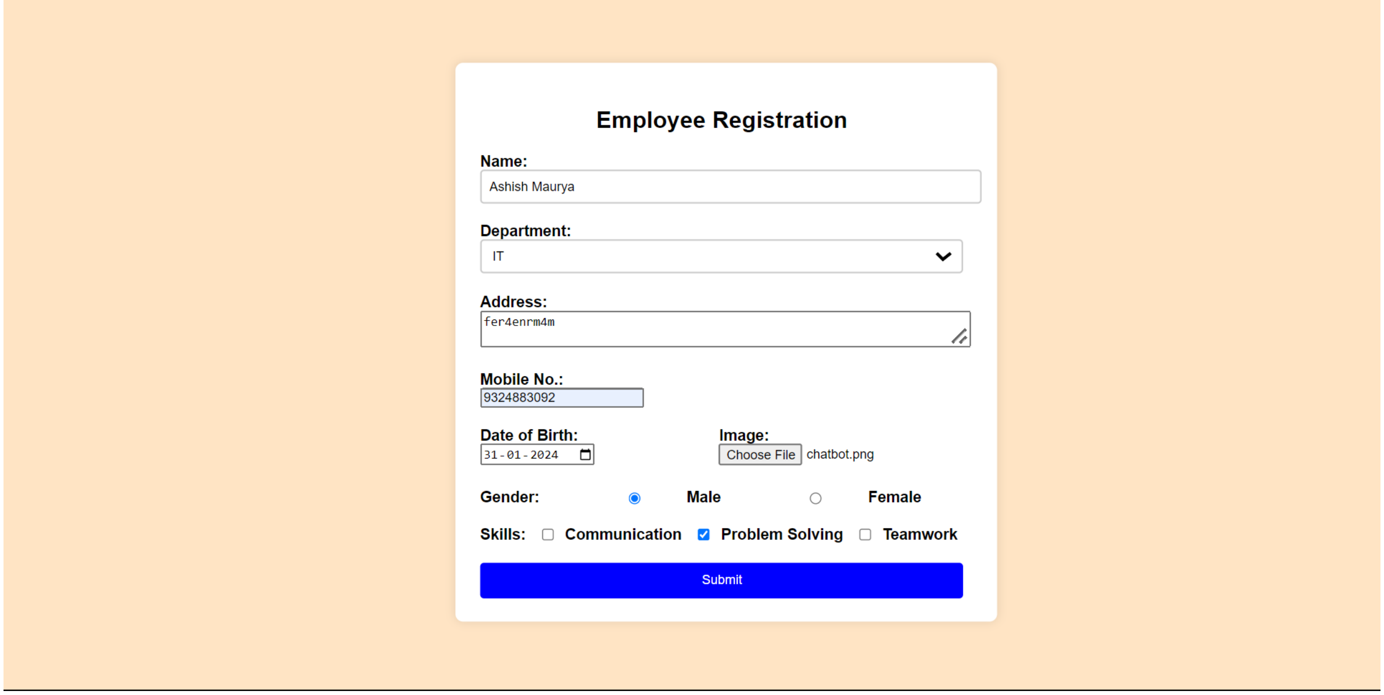
   f) Child Selector (>)
   Selects direct children of a specified element.
   Example: ul > li { list-style-type: square; }

   g) Adjacent Sibling Selector (+)
   Selects an element that is immediately preceded by a specified sibling element.
   Example: h2 + p { font-weight: bold; }

Conclusion: Thus we successfully learnt and implemented Inline, External and Embedded CSS in HTML.

# EXPERIMENT 3

a) Create employee registration webpage using HTML form objects. (include all the elements) also display on alert message box



**Employee Registration**

Name: Ashish Maurya

Department: IT

Address: fer4enrm4m

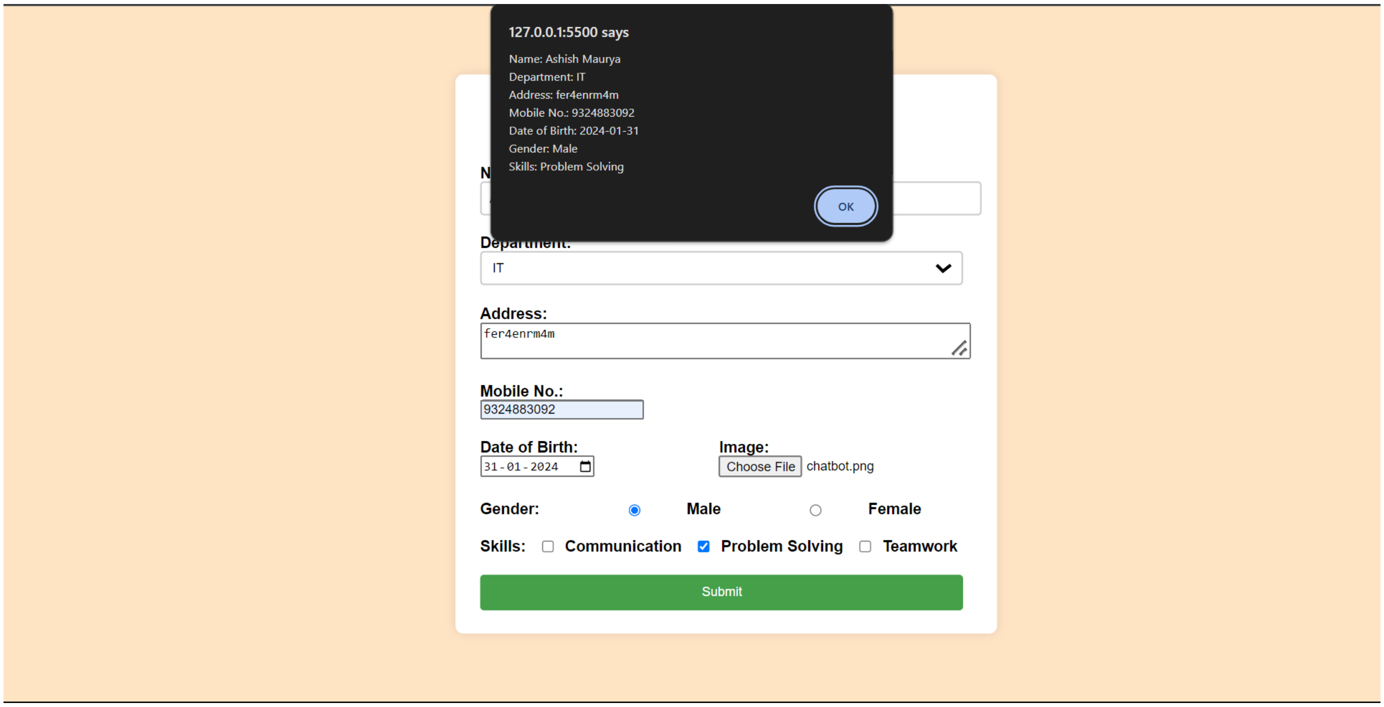
Mobile No.: 9324883092

Date of Birth: 31-01-2024

Image: Choose File chatbot.png

Gender: ● Male ○ Female

Skills: □ Communication ☑ Problem Solving □ Teamwork



127.0.0.1:5500 says

Name: Ashish Maurya
Department: IT
Address: fer4enrm4m
Mobile No.: 9324883092
Date of Birth: 2024-01-31
Gender: Male
Skills: Problem Solving

Department: IT

Address: fer4enrm4m

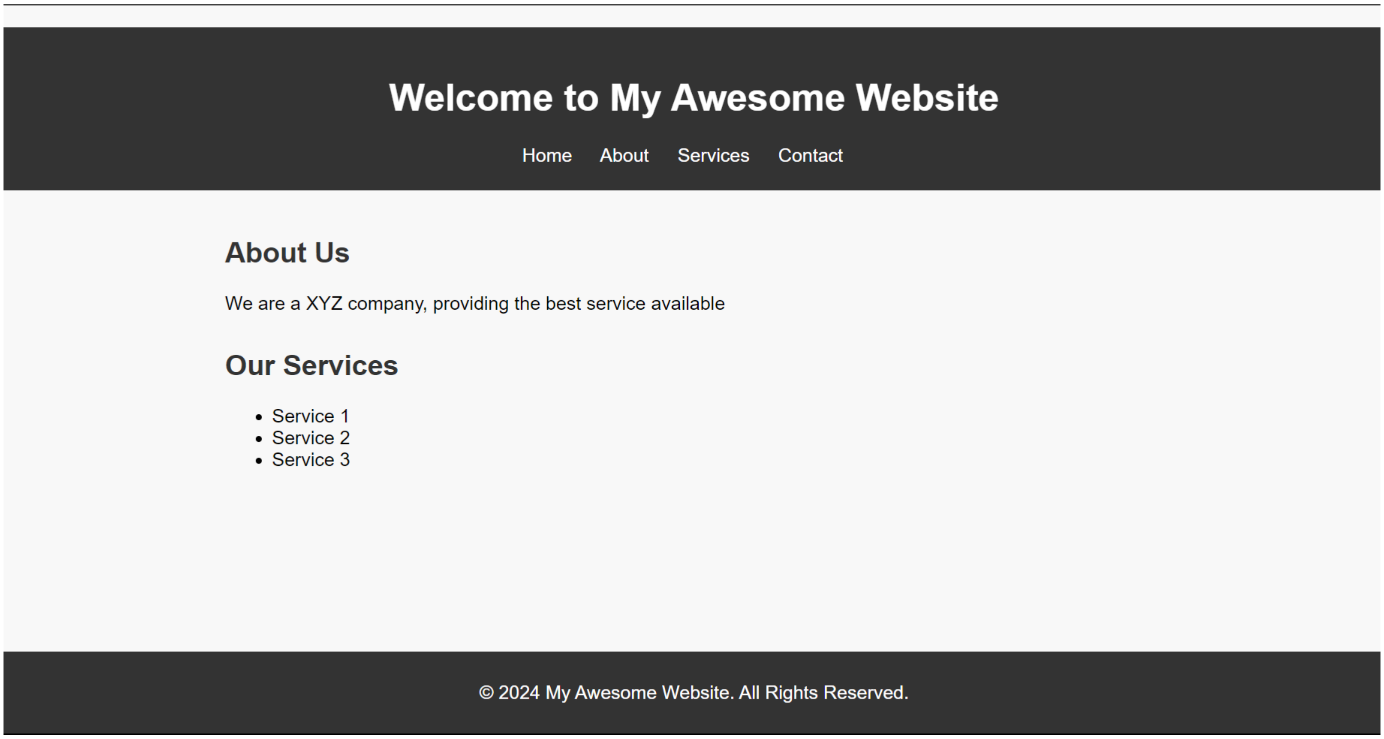
Mobile No.: 9324883092

Date of Birth: 31-01-2024

Image: Choose File chatbot.png

Gender: ● Male ○ Female

Skills: □ Communication ☑ Problem Solving □ Teamwork



Welcome to My Awesome Website

Home About Services Contact

About Us

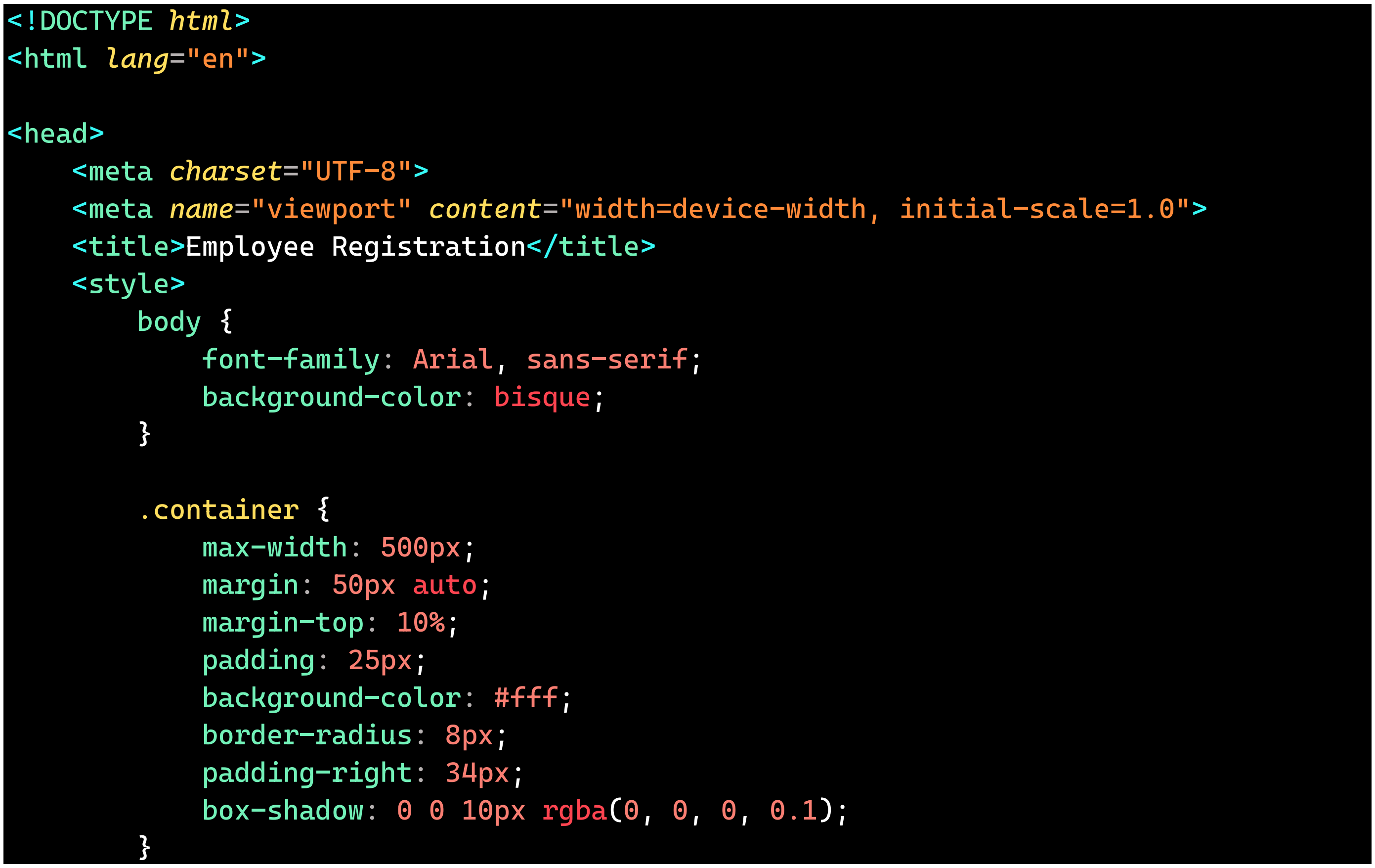
We are a XYZ company, providing the best service available

Our Services

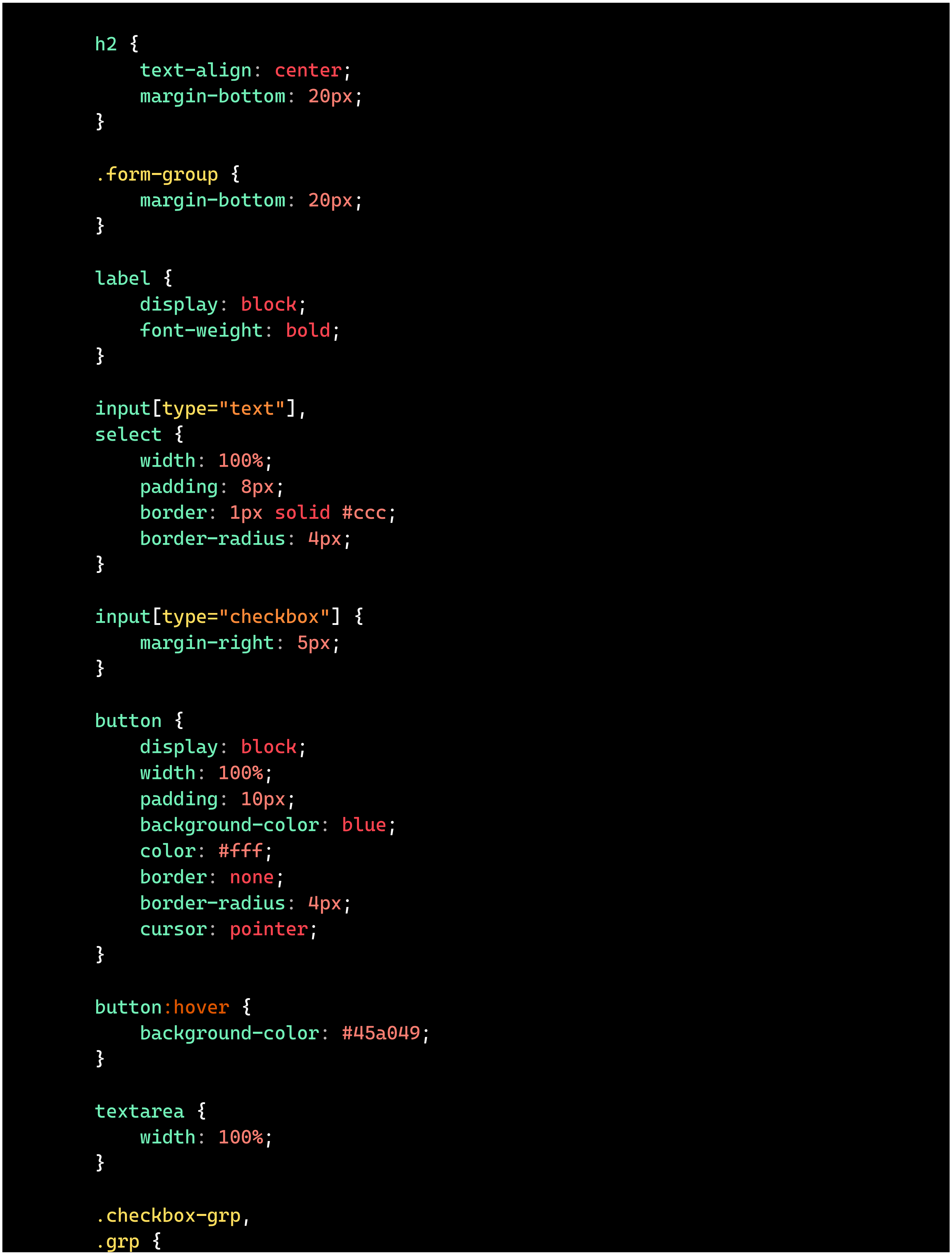
- Service 1
- Service 2
- Service 3

© 2024 My Awesome Website. All Rights Reserved.

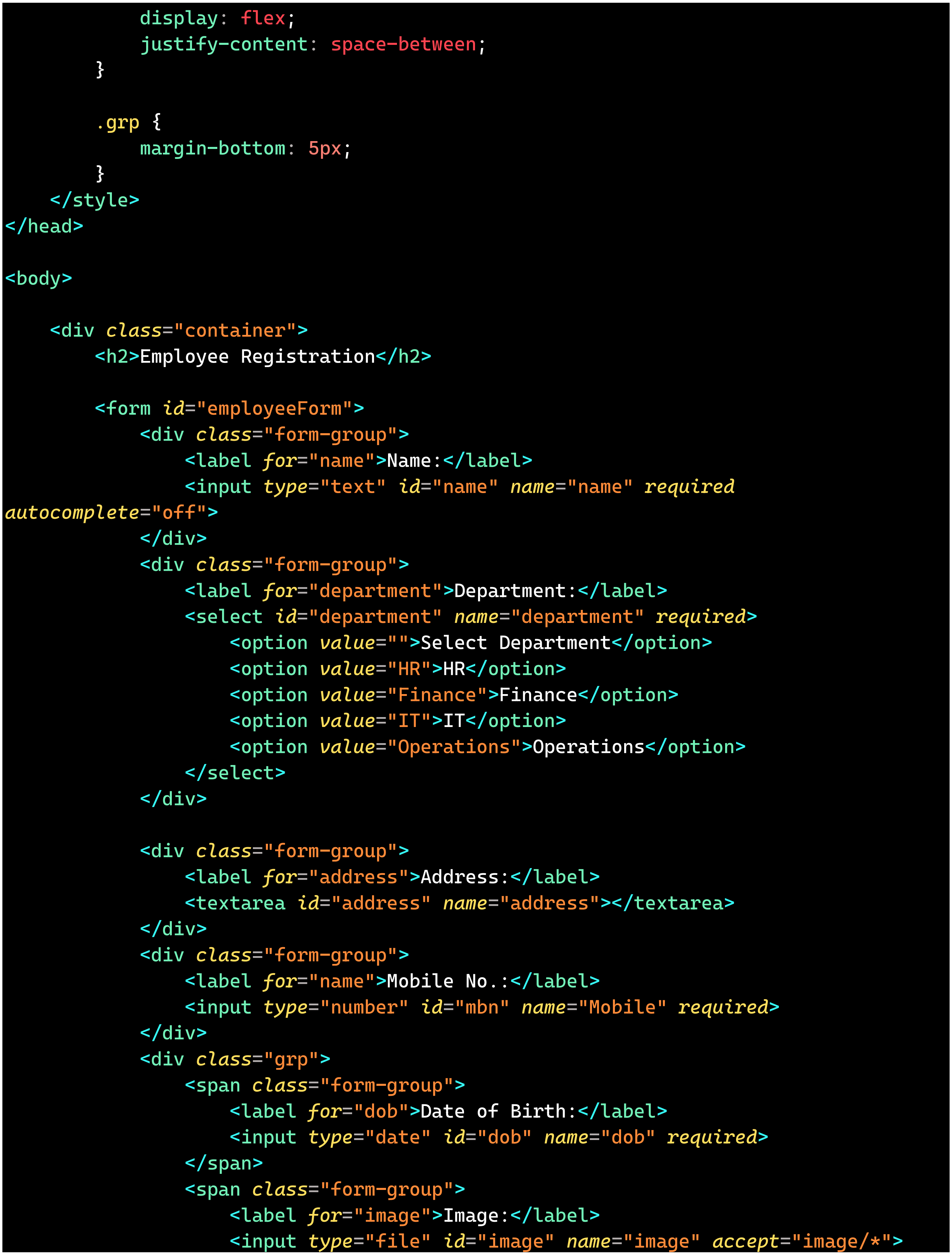
## A)



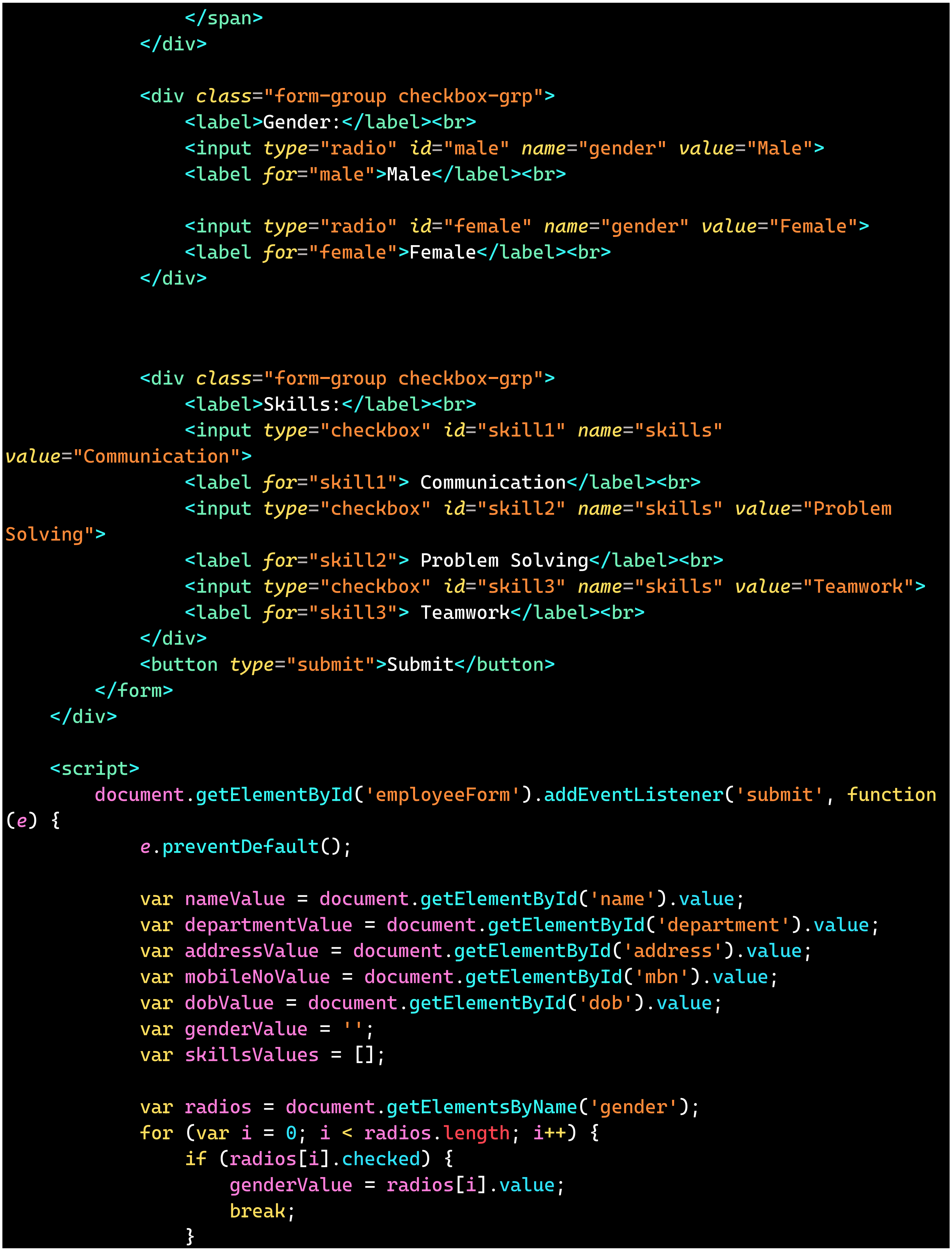
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Employee Registration</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: bisque;
        }

        .container {
            max-width: 500px;
            margin: 50px auto;
            margin-top: 10%;
            padding: 25px;
            background-color: #fff;
            border-radius: 8px;
            padding-right: 34px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
        }
```



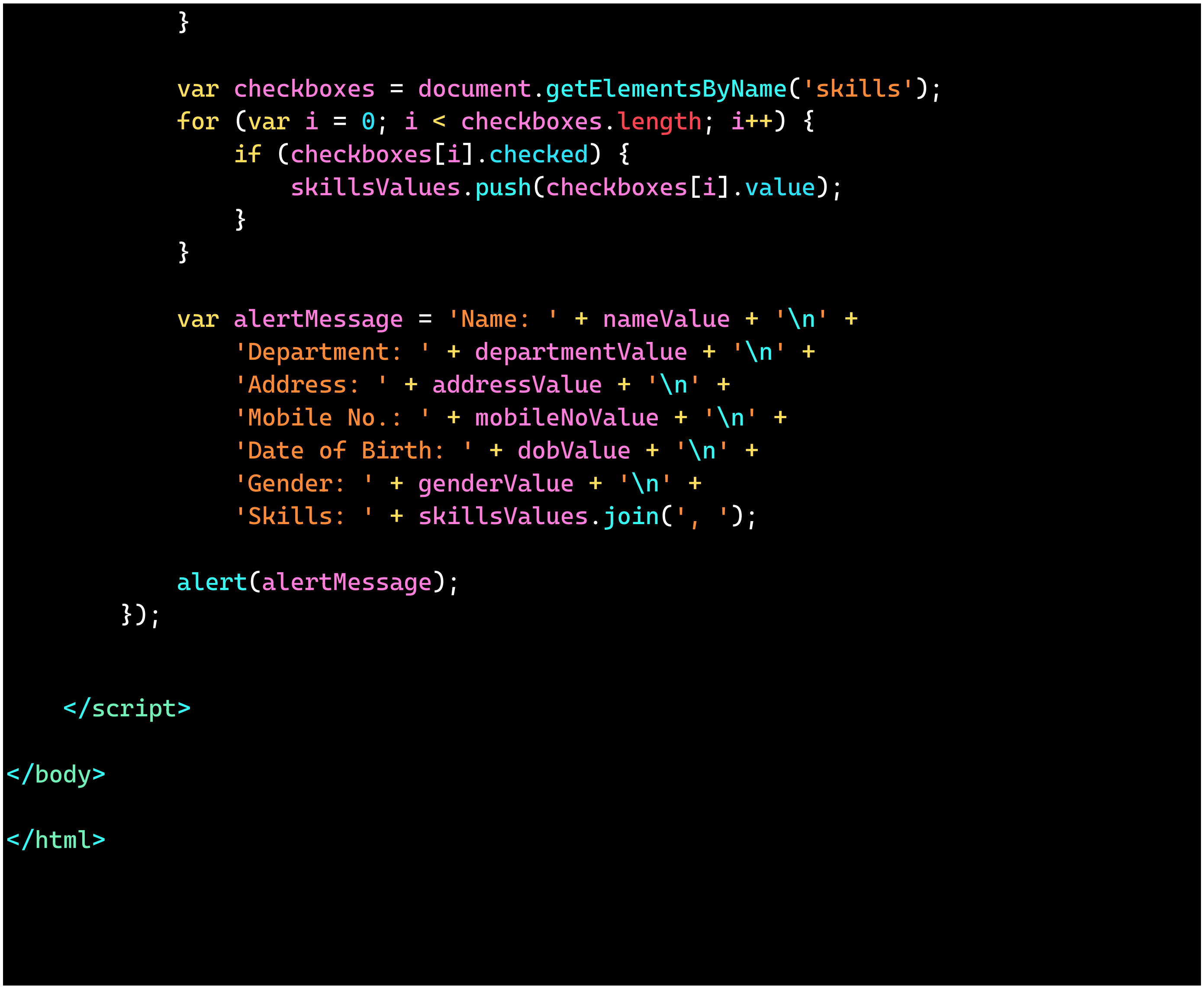
```
h2 {
    text-align: center;
    margin-bottom: 20px;
}

.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    font-weight: bold;
}

input[type="text"],
select {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 4px;
}

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

button {
    display: block;
    width: 100%;
    padding: 10px;
    background-color: blue;
    color: #fff;
    border: none;
    border-radius: 4px;
    cursor: pointer;
}

button:hover {
    background-color: #45a049;
}

textarea {
    width: 100%;
}

.checkbox-grp,
.grp {
```



```
            display: flex;
            justify-content: space-between;
        }

        .grp {
            margin-bottom: 5px;
        }
    </style>
</head>

<body>

    <div class="container">
        <h2>Employee Registration</h2>

        <form id="employeeForm">
            <div class="form-group">
                <label for="name">Name:</label>
                <input type="text" id="name" name="name" required
autocomplete="off">
            </div>
            <div class="form-group">
                <label for="department">Department:</label>
                <select id="department" name="department" required>
                    <option value="">Select Department</option>
                    <option value="HR">HR</option>
                    <option value="Finance">Finance</option>
                    <option value="IT">IT</option>
                    <option value="Operations">Operations</option>
                </select>
            </div>

            <div class="form-group">
                <label for="address">Address:</label>
                <textarea id="address" name="address"></textarea>
            </div>
            <div class="form-group">
                <label for="name">Mobile No.:</label>
                <input type="number" id="mbn" name="Mobile" required>
            </div>
            <div class="grp">
                <span class="form-group">
                    <label for="dob">Date of Birth:</label>
                    <input type="date" id="dob" name="dob" required>
                </span>
                <span class="form-group">
                    <label for="image">Image:</label>
                    <input type="file" id="image" name="image" accept="image/*">
```

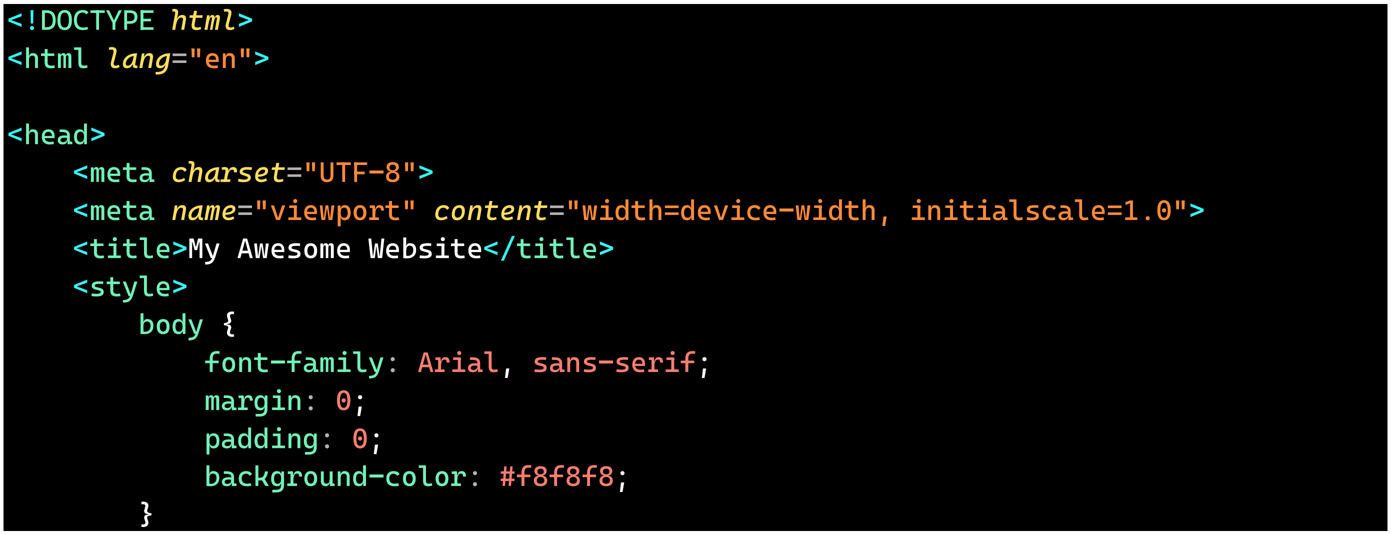


```
                </span>
            </div>

            <div class="form-group checkbox-grp">
                <label>Gender:</label><br>
                <input type="radio" id="male" name="gender" value="Male">
                <label for="male">Male</label><br>

                <input type="radio" id="female" name="gender" value="Female">
                <label for="female">Female</label><br>
            </div>



            <div class="form-group checkbox-grp">
                <label>Skills:</label><br>
                <input type="checkbox" id="skill1" name="skills"
value="Communication">
                <label for="skill1"> Communication</label><br>
                <input type="checkbox" id="skill2" name="skills" value="Problem
Solving">
                <label for="skill2"> Problem Solving</label><br>
                <input type="checkbox" id="skill3" name="skills" value="Teamwork">
                <label for="skill3"> Teamwork</label><br>
            </div>
            <button type="submit">Submit</button>
        </form>
    </div>

    <script>
        document.getElementById('employeeForm').addEventListener('submit', function
(e) {
            e.preventDefault();

            var nameValue = document.getElementById('name').value;
            var departmentValue = document.getElementById('department').value;
            var addressValue = document.getElementById('address').value;
            var mobileNoValue = document.getElementById('mbn').value;
            var dobValue = document.getElementById('dob').value;
            var genderValue = '';
            var skillsValues = [];

            var radios = document.getElementsByName('gender');
            for (var i = 0; i < radios.length; i++) {
                if (radios[i].checked) {
                    genderValue = radios[i].value;
                    break;
                }
```

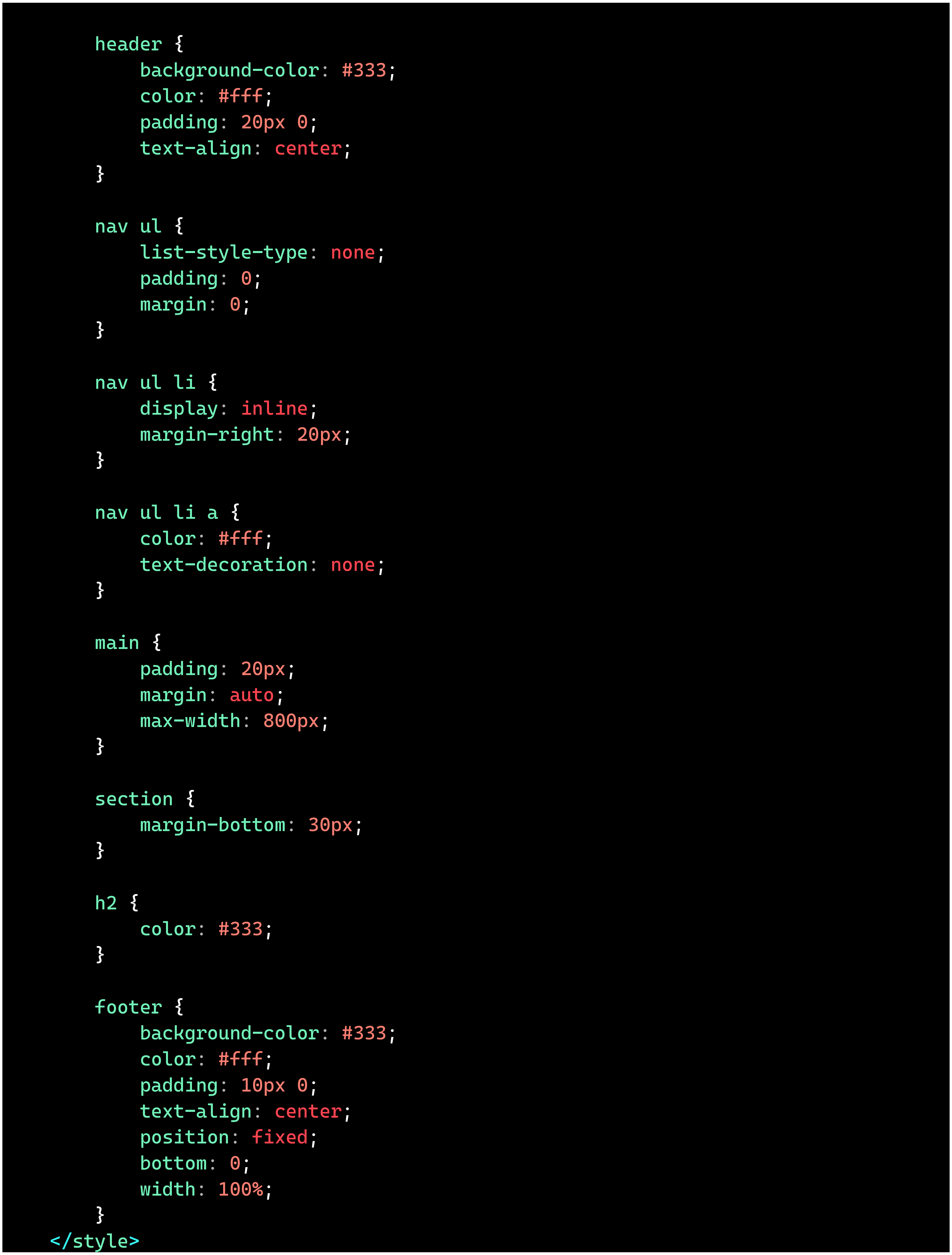


```
            }

            var checkboxes = document.getElementsByName('skills');
            for (var i = 0; i < checkboxes.length; i++) {
                if (checkboxes[i].checked) {
                    skillsValues.push(checkboxes[i].value);
                }
            }

            var alertMessage = 'Name: ' + nameValue + '\n' +
                'Department: ' + departmentValue + '\n' +
                'Address: ' + addressValue + '\n' +
                'Mobile No.: ' + mobileNoValue + '\n' +
                'Date of Birth: ' + dobValue + '\n' +
                'Gender: ' + genderValue + '\n' +
                'Skills: ' + skillsValues.join(', ');

            alert(alertMessage);
        });


    </script>

</body>

</html>
```

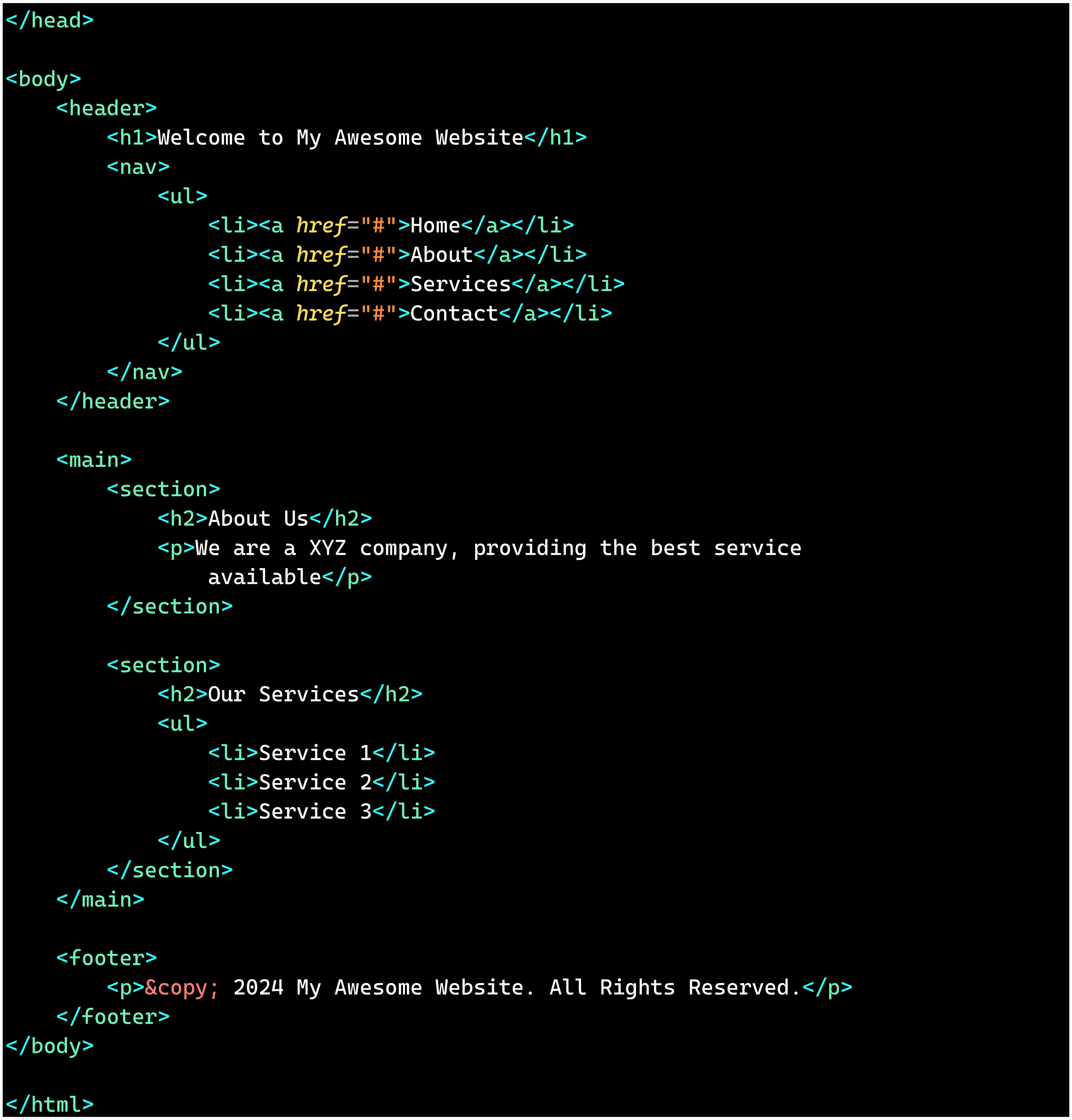
**B)**



```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initialscale=1.0">
    <title>My Awesome Website</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            margin: 0;
            padding: 0;
            background-color: #f8f8f8;
        }
```



```
        header {
            background-color: #333;
            color: #fff;
            padding: 20px 0;
            text-align: center;
        }

        nav ul {
            list-style-type: none;
            padding: 0;
            margin: 0;
        }

        nav ul li {
            display: inline;
            margin-right: 20px;
        }

        nav ul li a {
            color: #fff;
            text-decoration: none;
        }

        main {
            padding: 20px;
            margin: auto;
            max-width: 800px;
        }

        section {
            margin-bottom: 30px;
        }

        h2 {
            color: #333;
        }

        footer {
            background-color: #333;
            color: #fff;
            padding: 10px 0;
            text-align: center;
            position: fixed;
            bottom: 0;
            width: 100%;
        }
    </style>
```



```
</head>

<body>
    <header>
        <h1>Welcome to My Awesome Website</h1>
        <nav>
            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About</a></li>
                <li><a href="#">Services</a></li>
                <li><a href="#">Contact</a></li>
            </ul>
        </nav>
    </header>

    <main>
        <section>
            <h2>About Us</h2>
            <p>We are a XYZ company, providing the best service
                available</p>
        </section>

        <section>
            <h2>Our Services</h2>
            <ul>
                <li>Service 1</li>
                <li>Service 2</li>
                <li>Service 3</li>
            </ul>
        </section>
    </main>

    <footer>
        <p>&copy; 2024 My Awesome Website. All Rights Reserved.</p>
    </footer>
</body>

</html>
```