DEVELOPER'S PORTFOLIO

TASK - 4

INTRODUCTION TO DEVELOPER'S PORTFOLIO

Developers build portfolio websites as full-stack developer sample projects to showcase their skills and impress clients. As a student or professional learning web development, you must practice making portfolio websites to gain knowledge and experience in efficient front end web development technology.

Campus Name :- Ajeenkya DY Patil University Pune

Name :- Devendra Singh

Batch :- Bachelor's In Computer Application (BCA)

Year :- 1st year

URN :- 2022-B-09072004C

ABOUT TASK - 4

Integrate the APIs to Frontend to ensure the Dynamic features of websites.

- Point base API to the servers base URL.
- Design APIcalls for each element.
- Handle errors in the output.
- Render output of APIs to different low level components.
- Secure content of post APIs.

Introduction to HTML

HTML stands for Hypertext Markup Language and is the standard language used for creating web pages. It is a markup language that uses tags to define elements such as headings, paragraphs, links, and images.

HTML provides the structure of a web page and allows developers to create content that can be displayed on different devices and browsers. It is the foundation of every website and is essential for creating web pages.

Introduction to CSS

CSS stands for Cascading Style Sheets and is used to style web pages created with HTML. It allows developers to control the layout, font, color, and other visual aspects of a web page.

CSS works by selecting HTML elements and applying styles to them. It can be used to create responsive designs that adapt to different screen sizes and devices, making it an essential tool for creating modern web pages.

Introduction to JS

JavaScript is a programming language used to add interactivity and dynamic behavior to web pages. It can be used to create animations, validate forms, and interact with APIs and databases.

JavaScript works by manipulating the Document Object Model (DOM), which represents the structure of an HTML document. It can access and modify elements on a web page, respond to user events such as clicks and key presses, and communicate with servers to retrieve or send data.

EXTENTIONS USED

Live Server

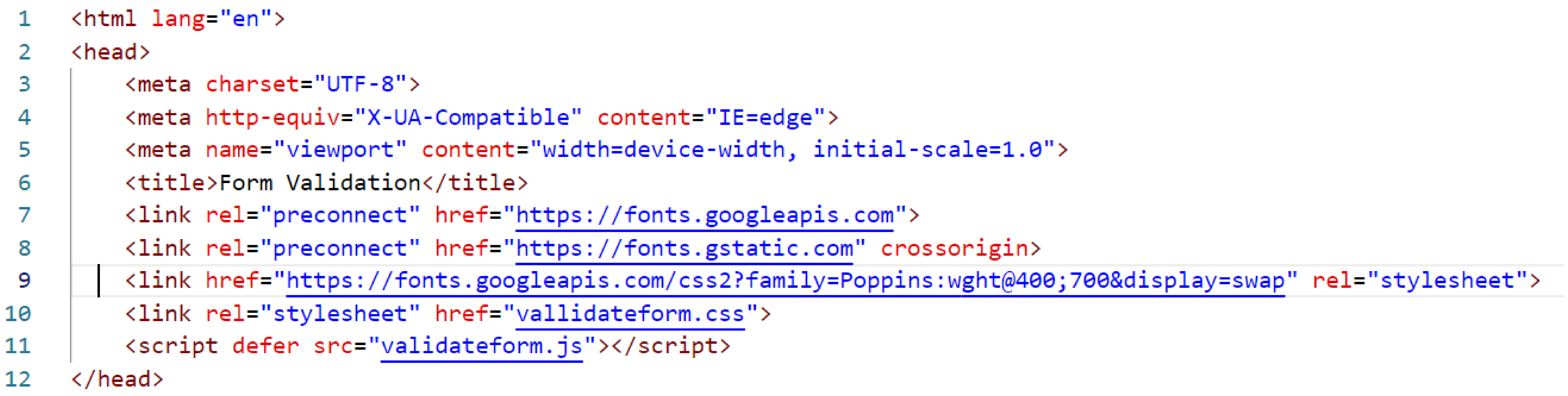
JavaScript (ES6) code snippets

Code Runner

APPLICATION USED

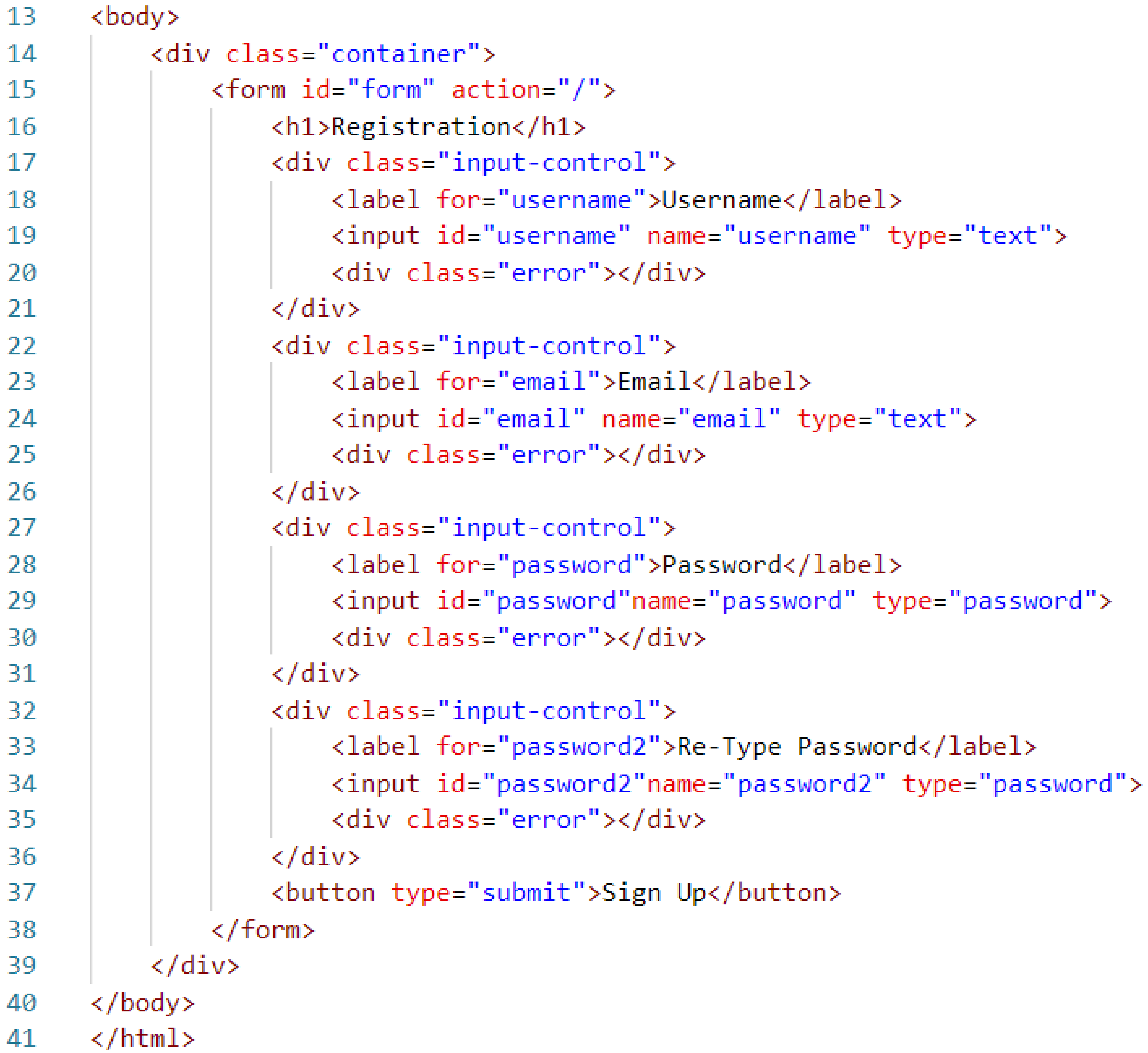
Visual Studio Code

CODE



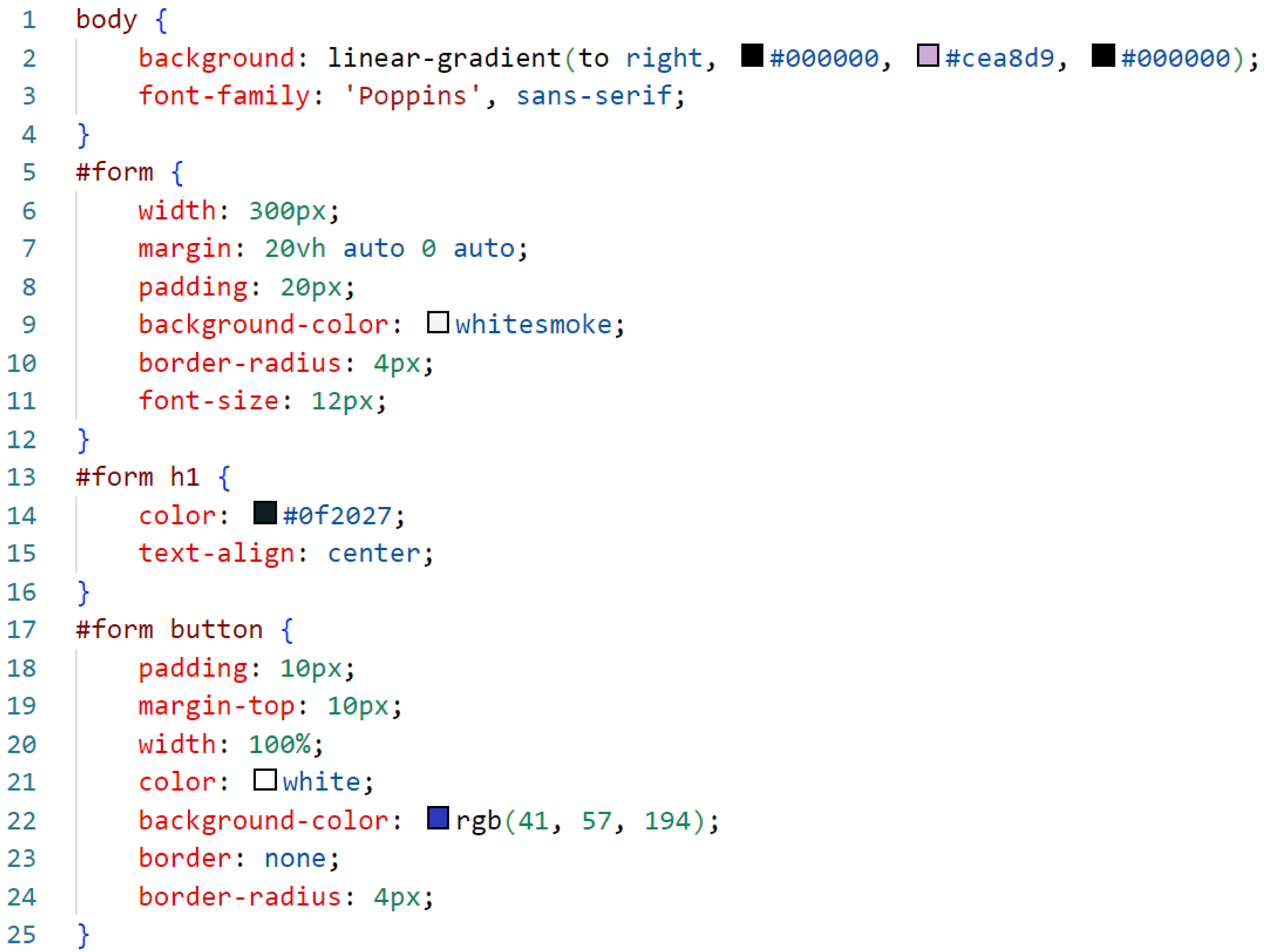
```html
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Form Validation</title>
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;700&display=swap" rel="stylesheet">
    <link rel="stylesheet" href="vallidateform.css">
    <script defer src="validateform.js"></script>
</head>
```

CODE



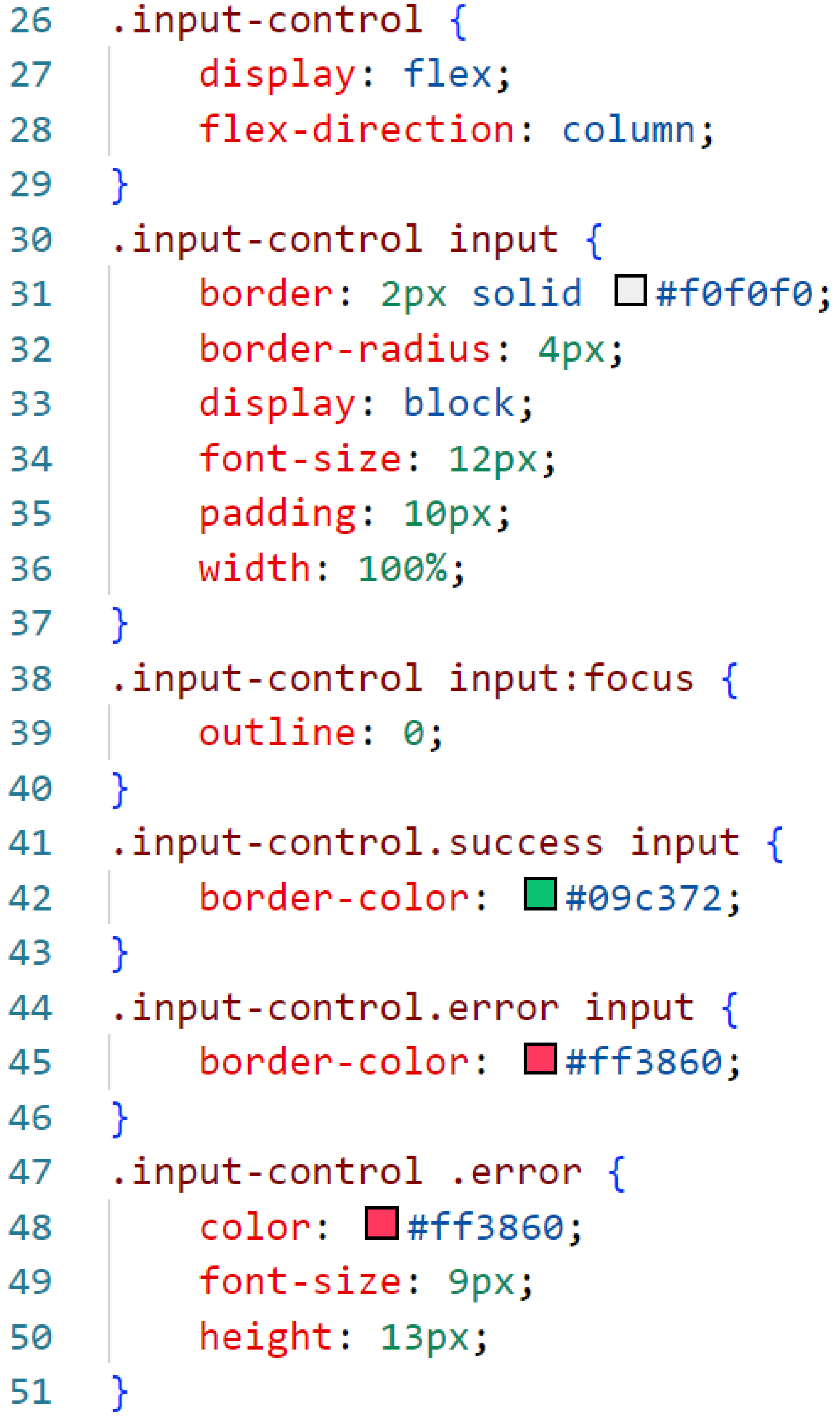
```html
<body>
    <div class="container">
        <form id="form" action="/">
            <h1>Registration</h1>
            <div class="input-control">
                <label for="username">Username</label>
                <input id="username" name="username" type="text">
                <div class="error"></div>
            </div>
            <div class="input-control">
                <label for="email">Email</label>
                <input id="email" name="email" type="text">
                <div class="error"></div>
            </div>
            <div class="input-control">
                <label for="password">Password</label>
                <input id="password"name="password" type="password">
                <div class="error"></div>
            </div>
            <div class="input-control">
                <label for="password2">Re-Type Password</label>
                <input id="password2"name="password2" type="password">
                <div class="error"></div>
            </div>
            <button type="submit">Sign Up</button>
        </form>
    </div>
</body>
</html>
```

CSS CODE



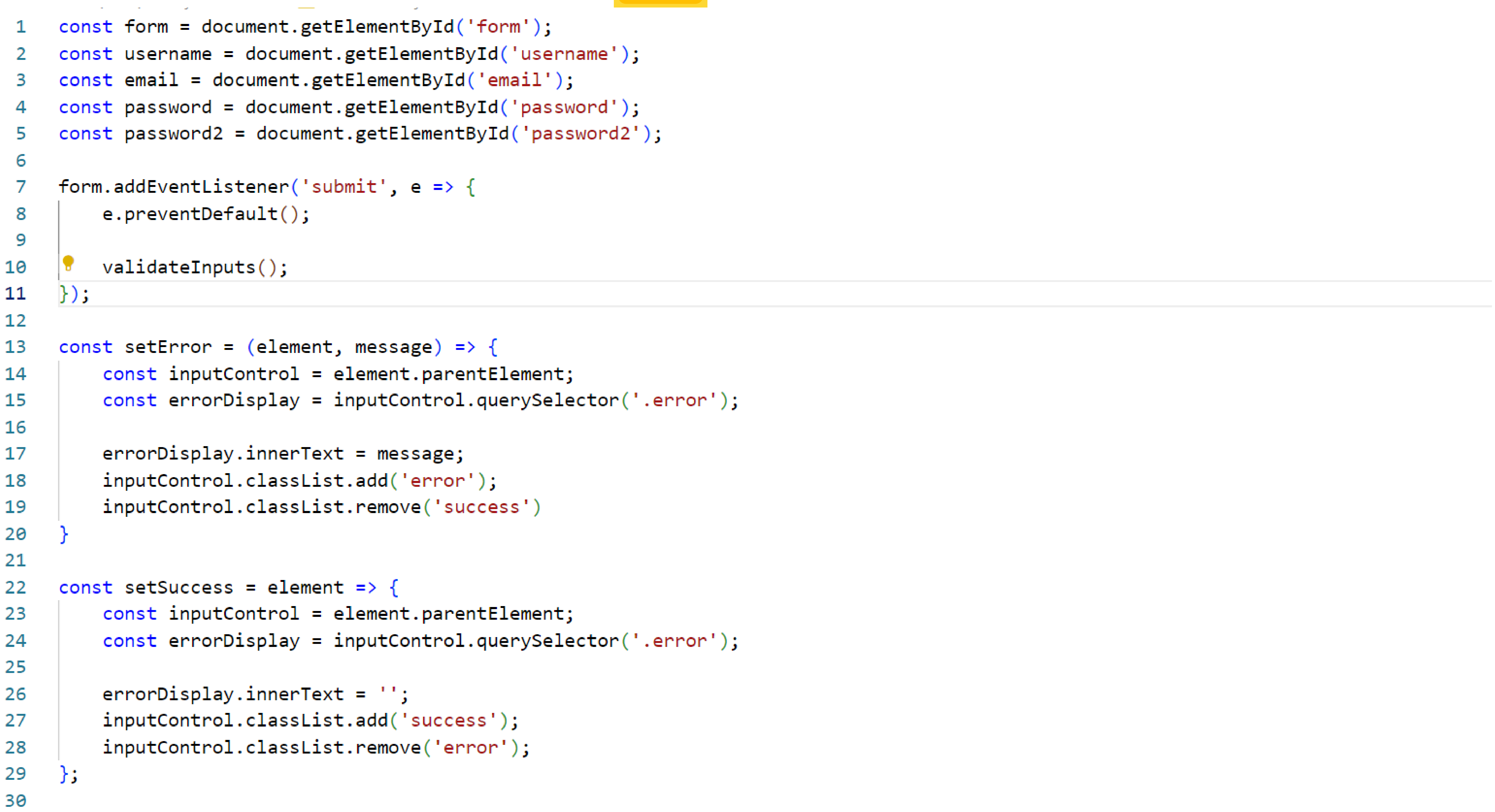
```css
body {
    background: linear-gradient(to right, #000000, #cea8d9, #000000);
    font-family: 'Poppins', sans-serif;
}
#form {
    width: 300px;
    margin: 20vh auto 0 auto;
    padding: 20px;
    background-color: whitesmoke;
    border-radius: 4px;
    font-size: 12px;
}
#form h1 {
    color: #0f2027;
    text-align: center;
}
#form button {
    padding: 10px;
    margin-top: 10px;
    width: 100%;
    color: white;
    background-color: rgb(41, 57, 194);
    border: none;
    border-radius: 4px;
}
```

CODE



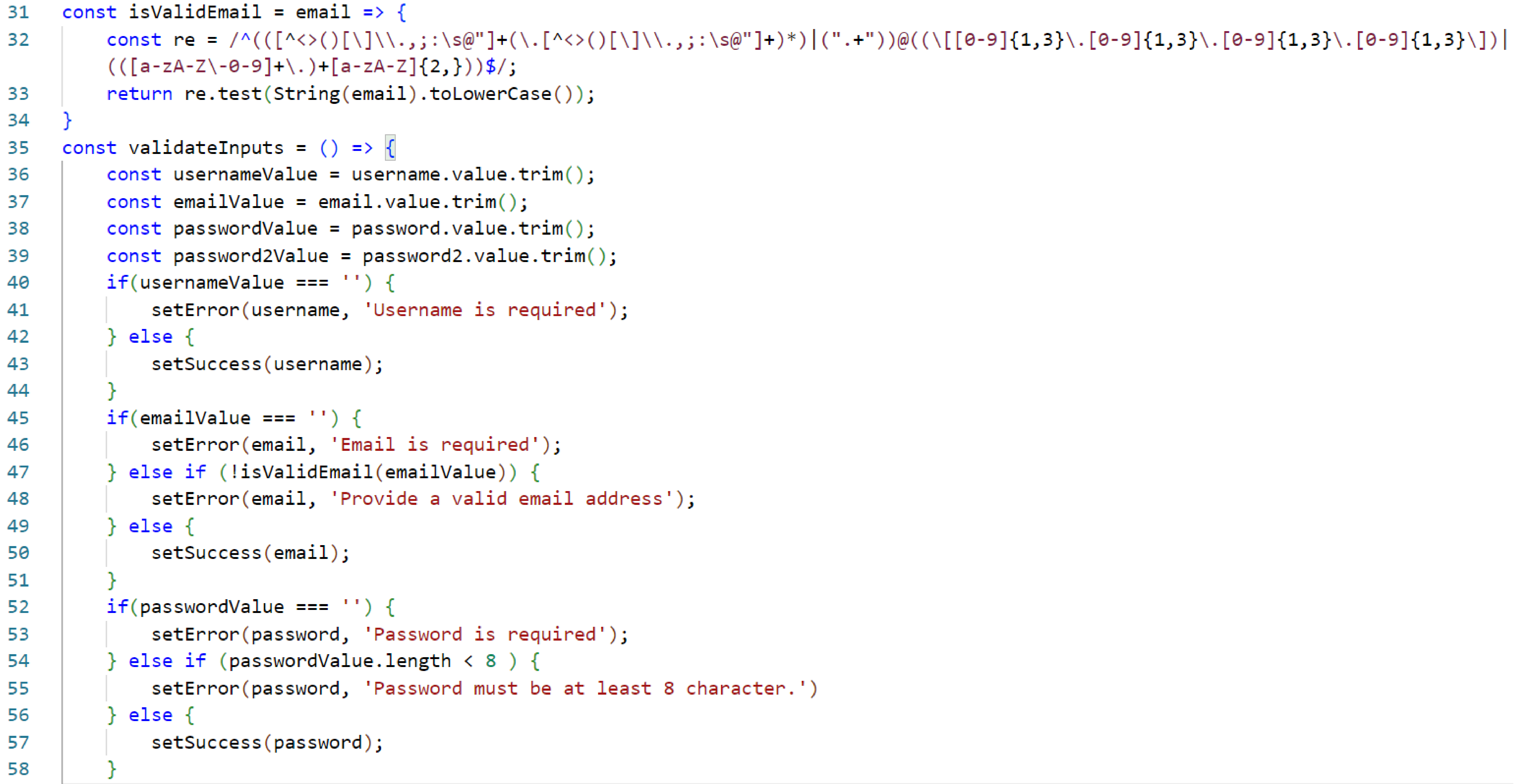
```css
.input-control {
    display: flex;
    flex-direction: column;
}
.input-control input {
    border: 2px solid □#f0f0f0;
    border-radius: 4px;
    display: block;
    font-size: 12px;
    padding: 10px;
    width: 100%;
}
.input-control input:focus {
    outline: 0;
}
.input-control.success input {
    border-color: 🟩#09c372;
}
.input-control.error input {
    border-color: 🟥#ff3860;
}
.input-control .error {
    color: 🟥#ff3860;
    font-size: 9px;
    height: 13px;
}
```

CODE



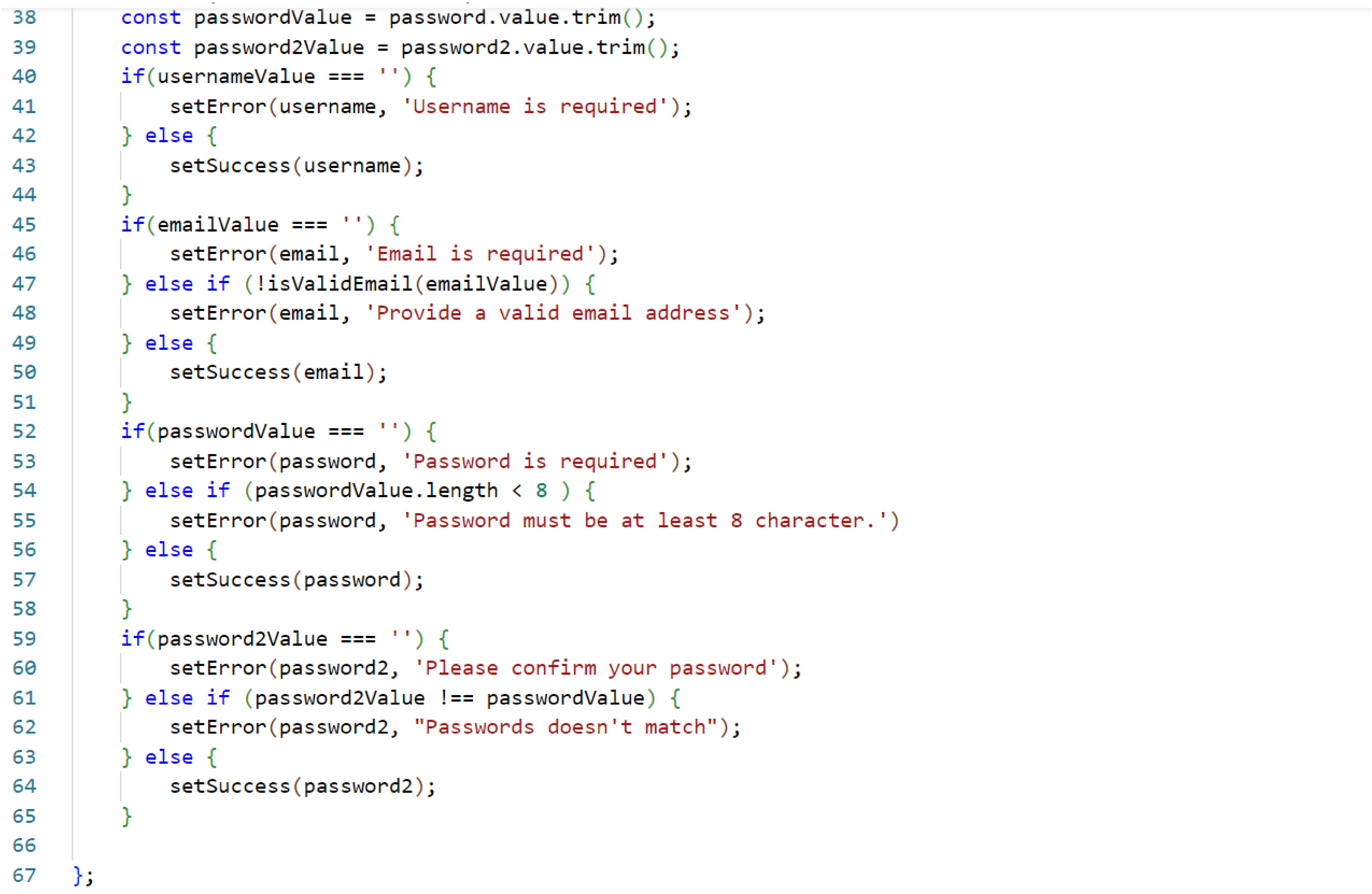
```javascript
const form = document.getElementById('form');
const username = document.getElementById('username');
const email = document.getElementById('email');
const password = document.getElementById('password');
const password2 = document.getElementById('password2');

form.addEventListener('submit', e => {
    e.preventDefault();

    validateInputs();
});

const setError = (element, message) => {
    const inputControl = element.parentElement;
    const errorDisplay = inputControl.querySelector('.error');

    errorDisplay.innerText = message;
    inputControl.classList.add('error');
    inputControl.classList.remove('success')
}

const setSuccess = element => {
    const inputControl = element.parentElement;
    const errorDisplay = inputControl.querySelector('.error');

    errorDisplay.innerText = '';
    inputControl.classList.add('success');
    inputControl.classList.remove('error');
};

```

CODE



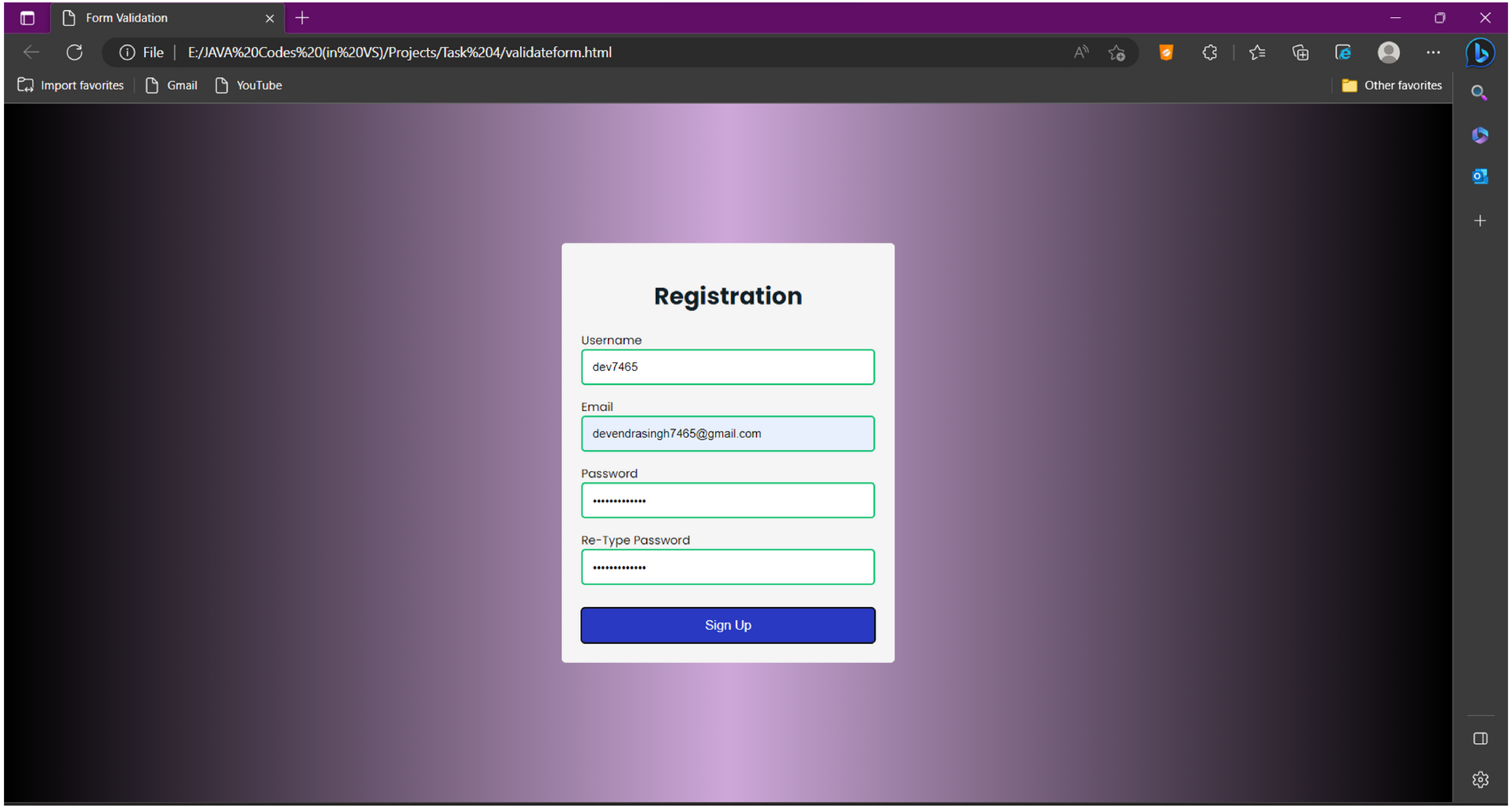
```
const isValidEmail = email => {
    const re = /^(([^<>()[\]\\.,;:\s@"]+(\.[^<>()[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/;
    return re.test(String(email).toLowerCase());
}
const validateInputs = () => {
    const usernameValue = username.value.trim();
    const emailValue = email.value.trim();
    const passwordValue = password.value.trim();
    const password2Value = password2.value.trim();
    if(usernameValue === '') {
        setError(username, 'Username is required');
    } else {
        setSuccess(username);
    }
    if(emailValue === '') {
        setError(email, 'Email is required');
    } else if (!isValidEmail(emailValue)) {
        setError(email, 'Provide a valid email address');
    } else {
        setSuccess(email);
    }
    if(passwordValue === '') {
        setError(password, 'Password is required');
    } else if (passwordValue.length < 8 ) {
        setError(password, 'Password must be at least 8 character.')
    } else {
        setSuccess(password);
    }
```

CODE



```javascript
    const passwordValue = password.value.trim();
    const password2Value = password2.value.trim();
    if(usernameValue === '') {
        setError(username, 'Username is required');
    } else {
        setSuccess(username);
    }
    if(emailValue === '') {
        setError(email, 'Email is required');
    } else if (!isValidEmail(emailValue)) {
        setError(email, 'Provide a valid email address');
    } else {
        setSuccess(email);
    }
    if(passwordValue === '') {
        setError(password, 'Password is required');
    } else if (passwordValue.length < 8 ) {
        setError(password, 'Password must be at least 8 character.')
    } else {
        setSuccess(password);
    }
    if(password2Value === '') {
        setError(password2, 'Please confirm your password');
    } else if (password2Value !== passwordValue) {
        setError(password2, "Passwords doesn't match");
    } else {
        setSuccess(password2);
    }

};
```

AFTER RUNNING HTML CSS AND JS CODE



Form Validation
File | E:/JAVA%20Codes%20(in%20VS)/Projects/Task%204/validateform.html
Import favorites
Gmail
YouTube
Other favorites

Registration

Username
dev7465

Email
devendrasingh7465@gmail.com

Password
•••••••••••••

Re-Type Password
•••••••••••••

CONCLUSION

HTML, CSS, and JavaScript are the building blocks of modern web development. By understanding these technologies and following best practices, developers can create engaging, responsive, and secure web pages that meet the needs of users on different devices and platforms.

Whether you are a beginner or an experienced developer, there is always more to learn about web development. By staying curious and exploring new technologies and techniques, you can continue to improve your skills and create better web experiences for users.

GitHub Link - Click Here

THANK
YOU