

DEVELOPMENT ASSMENT – 4

CASE STUDY ON FORM VALIDATOR

Aim of the Case Study: - Practice and execute HTML, CSS & JS and create a Signup page for Netflix and validate the form

CODE: -

Index.html

```
<!DOCTYPE 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">
  <link rel="stylesheet" href="style.css">
  <title>Form validator</title>
</head>
<body>
  <form id="form" action="/">
    <div class="card">
      <h1>REGISTER NOW</h1>
      <div class="row">
        <label for="username">USERNAME</label>
        <input type="text" id="username" name="username" placeholder="Username" required>
        <div class="error"></div>
      </div>
      <div class="row">
        <label for="email">EMAIL</label>
        <input type="email" id="email" name="email" placeholder="Enter your email" >
      </div>
    </div>
  </form>
</body>
</html>
```

```

<div class="error"></div>

</div>

<div class="row">
  <label for="password">PASSWORD</label>
  <input type="password" id="password" name="password" placeholder="Password">
  <div class="error"></div>
</div>

<div class="row">
  <button><a href="index1.html">SIGN UP</a></button>
</div>
</form>
</div>
</body>
</html>

```

Style.css

```

@import url('https://fonts.googleapis.com/css2?family=Montserrat&display=swap');

*{
  margin:0;
  padding: 0;
  box-sizing: border-box;
}

body{
  height: 100vh;
  font-family: 'Montserrat', sans-serif;
  background-image: url(/netflix.jpg);
  font-weight: bold
}

.card{
  width:30%;
  background-color: rgb(0, 0, 0,0.9);
  padding:20px;

```

```
    margin:4em auto;

    border-radius: 5px;

}

label{

    display: block;

    color: #ce1616;

}

input{

    width: calc(100% - 30px);

    margin-top:1em;

    padding:0.5em;

    border-radius: 4px;

    font-family: inherit;

    border:2px solid #b11212;

}

input:focus{

    outline: 0;

}

.row{

    margin:1em;

}

h1{

    margin:10px;

    color: #cf1414;

}

.row .error{

    font-size: 0.7em;

    color:rgb(123, 21, 21);

    height: 12px;
```

```

}

.row.error input{
    border-color: rgb(176, 18, 18);
}

.row.success input{
    border-color:rgb(0, 0, 0);
}

button{
    width:calc(100% - 100px);
    background-color:#aa1717 ;
    color:white;
    font-family: inherit;
    font-weight: bold;
    font-size: 105%;
    padding: 0.5em 1em;
    border:none;
    border-radius: 4px;
    margin-top:1.5em;
    margin-left:2.5em;
    cursor: pointer;
    transition: 1s all ease-in-out ;
}

button:hover{
    background-color:#eeebeb ;
    box-shadow: 3px 3px 3px 1px #d1cece ;
    transform: translateY(-10%);
}

@media screen and (max-width:880px)
{
    .card{
        width:80%;
    }
}

```

```
    }  
  }  
  a{  
    color: #000000;  
  }
```

Script.js

```
const form=document.getElementById('form');  
const email=document.getElementById('email');  
const username=document.getElementById('username');  
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');
};

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-Z0-9)+\.)+[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');
    }

```

OUTPUT: -

