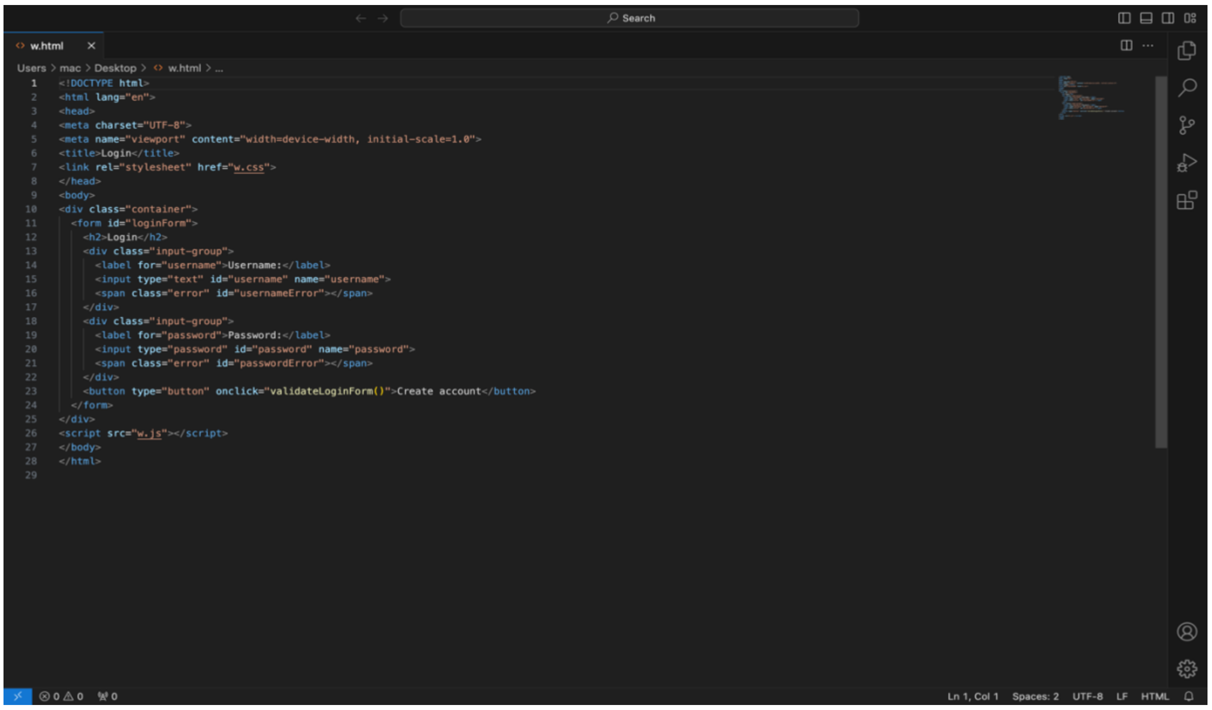
Zaynab Mohamud

669382

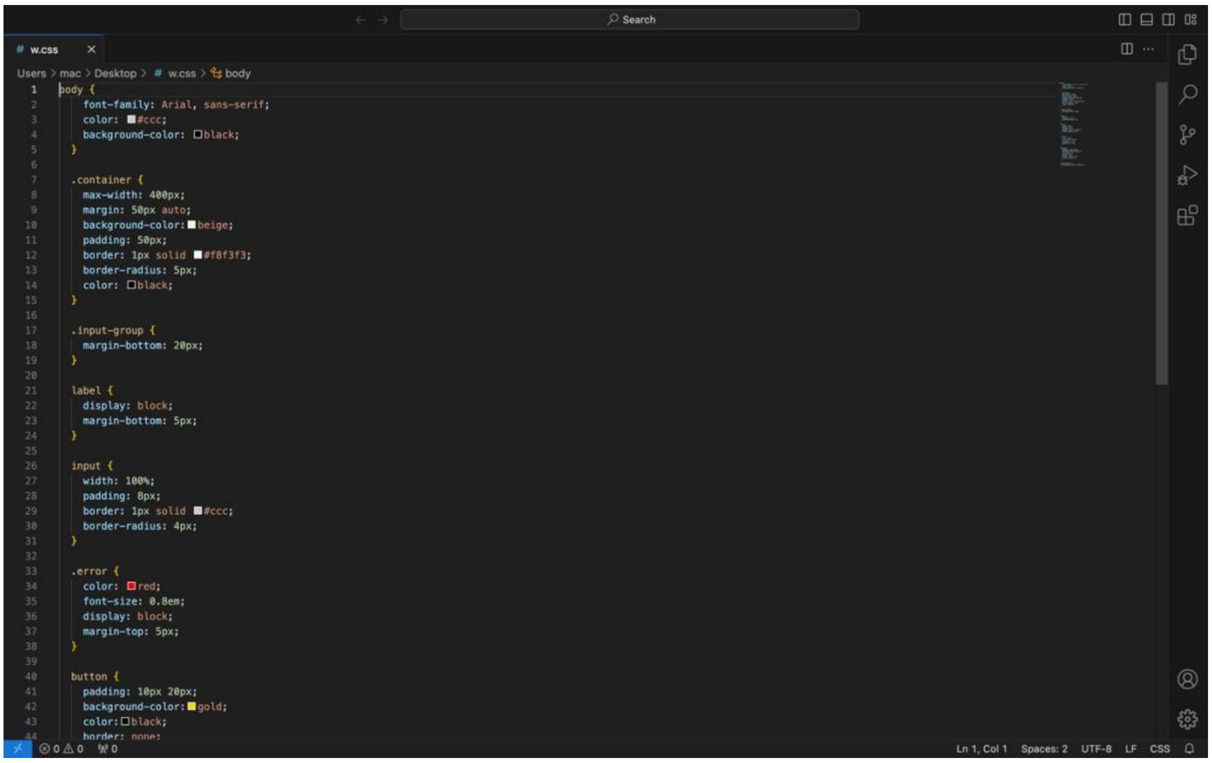
Procedural programming

Procedural HTML



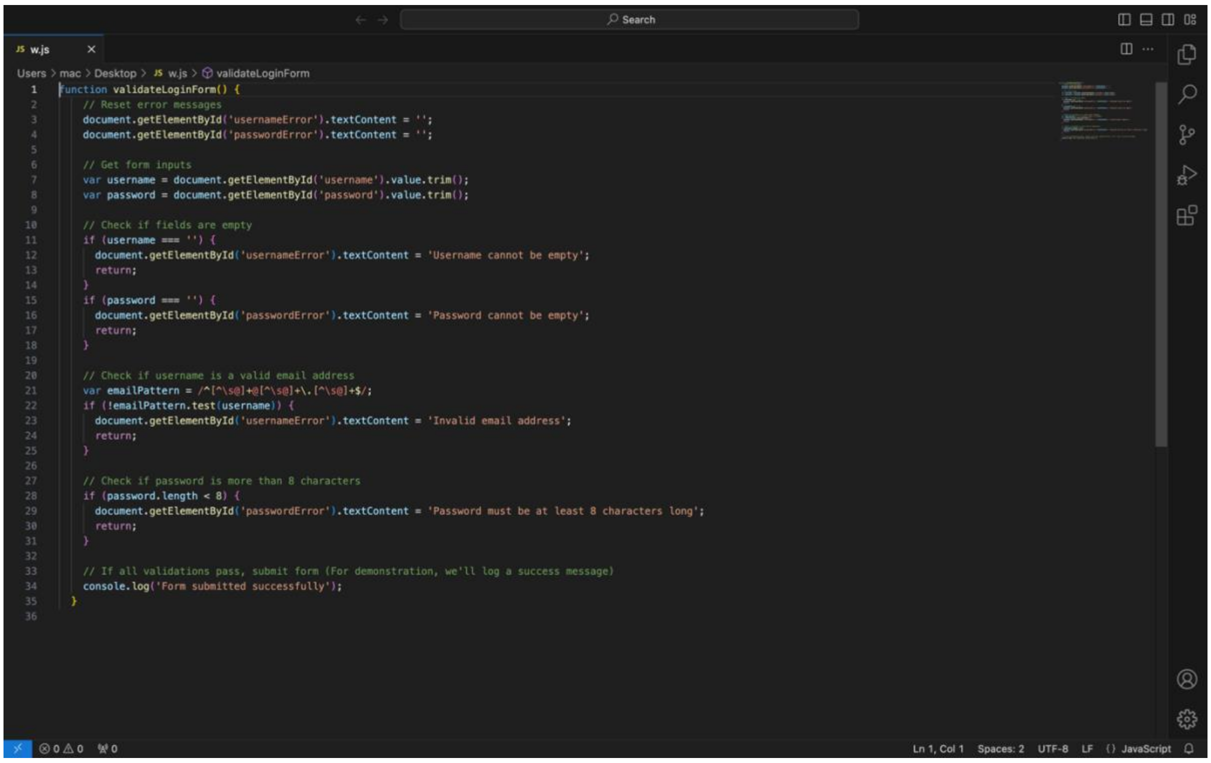
```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login</title>
<link rel="stylesheet" href="w.css">
</head>
<body>
<div class="container">
  <form id="loginForm">
    <h2>Login</h2>
    <div class="input-group">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username">
      <span class="error" id="usernameError"></span>
    </div>
    <div class="input-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password">
      <span class="error" id="passwordError"></span>
    </div>
    <button type="button" onclick="validateLoginForm()">Create account</button>
  </form>
</div>
<script src="w.js"></script>
</body>
</html>
```

ProceduralCSS



```css
body {
  font-family: Arial, sans-serif;
  color: #ccc;
  background-color: black;
}

.container {
  max-width: 400px;
  margin: 50px auto;
  background-color: beige;
  padding: 50px;
  border: 1px solid #f8f3f3;
  border-radius: 5px;
  color: black;
}

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

label {
  display: block;
  margin-bottom: 5px;
}

input {
  width: 100%;
  padding: 8px;
  border: 1px solid #ccc;
  border-radius: 4px;
}

.error {
  color: red;
  font-size: 0.8em;
  display: block;
  margin-top: 5px;
}

button {
  padding: 10px 20px;
  background-color: gold;
  color: black;
  border: none;
```

Procedural JS

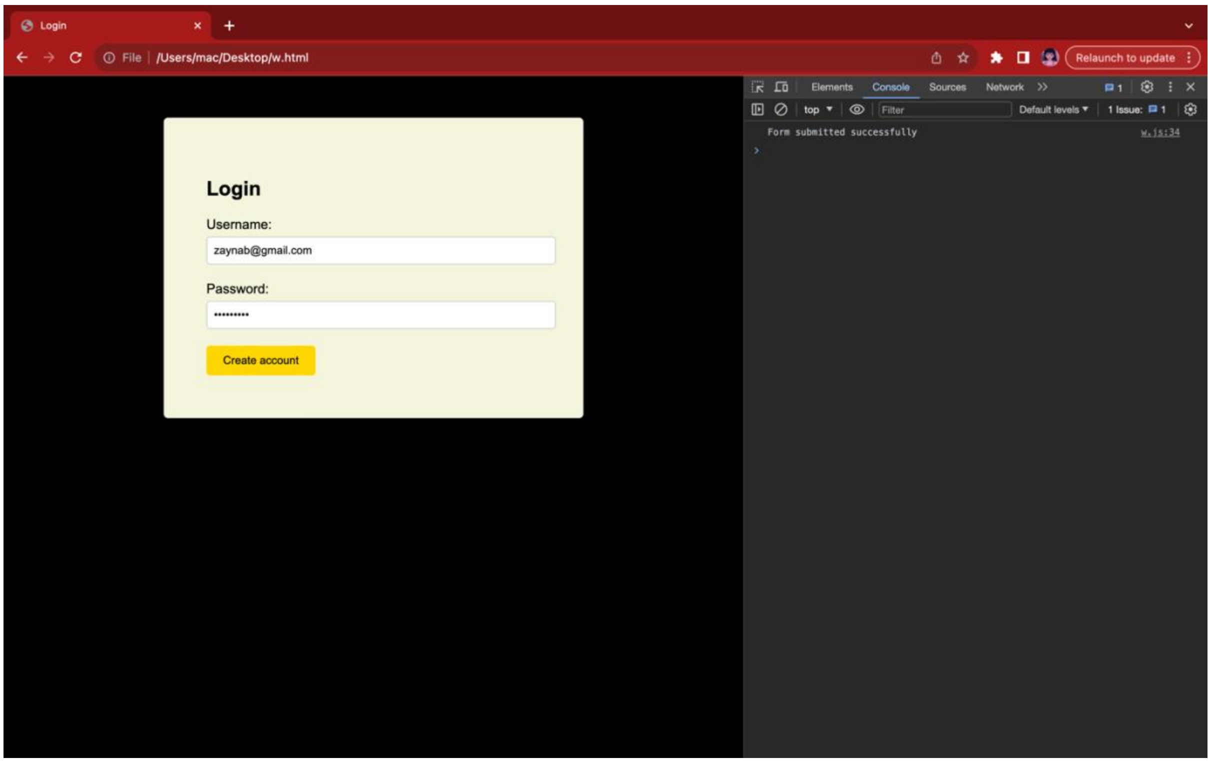


Users > mac > Desktop > w.js > validateLoginForm

```javascript
function validateLoginForm() {
  // Reset error messages
  document.getElementById('usernameError').textContent = '';
  document.getElementById('passwordError').textContent = '';

  // Get form inputs
  var username = document.getElementById('username').value.trim();
  var password = document.getElementById('password').value.trim();

  // Check if fields are empty
  if (username === '') {
    document.getElementById('usernameError').textContent = 'Username cannot be empty';
    return;
  }
  if (password === '') {
    document.getElementById('passwordError').textContent = 'Password cannot be empty';
    return;
  }

  // Check if username is a valid email address
  var emailPattern = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;
  if (!emailPattern.test(username)) {
    document.getElementById('usernameError').textContent = 'Invalid email address';
    return;
  }

  // Check if password is more than 8 characters
  if (password.length < 8) {
    document.getElementById('passwordError').textContent = 'Password must be at least 8 characters long';
    return;
  }

  // If all validations pass, submit form (For demonstration, we'll log a success message)
  console.log('Form submitted successfully');
}
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF {} JavaScript

Procedural Console



File | /Users/mac/Desktop/w.html

Login

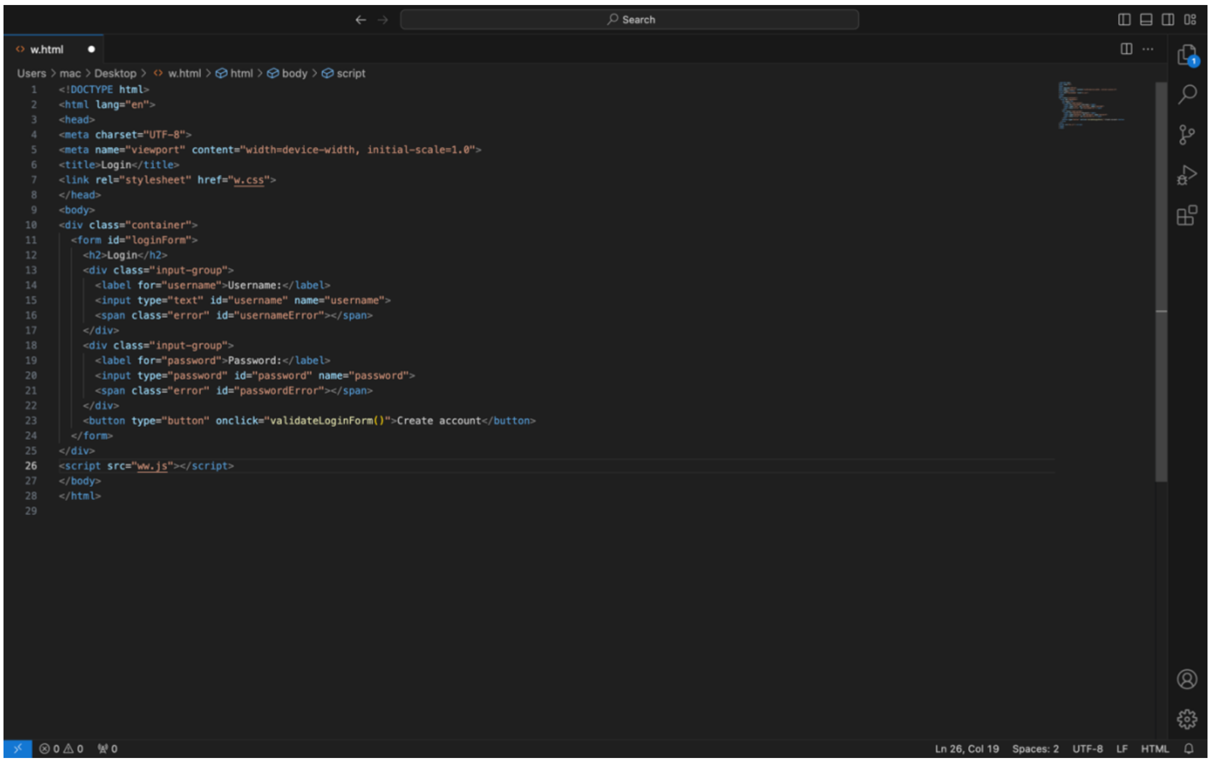
Username: zaynab@gmail.com

Password: •••••••••

Create account

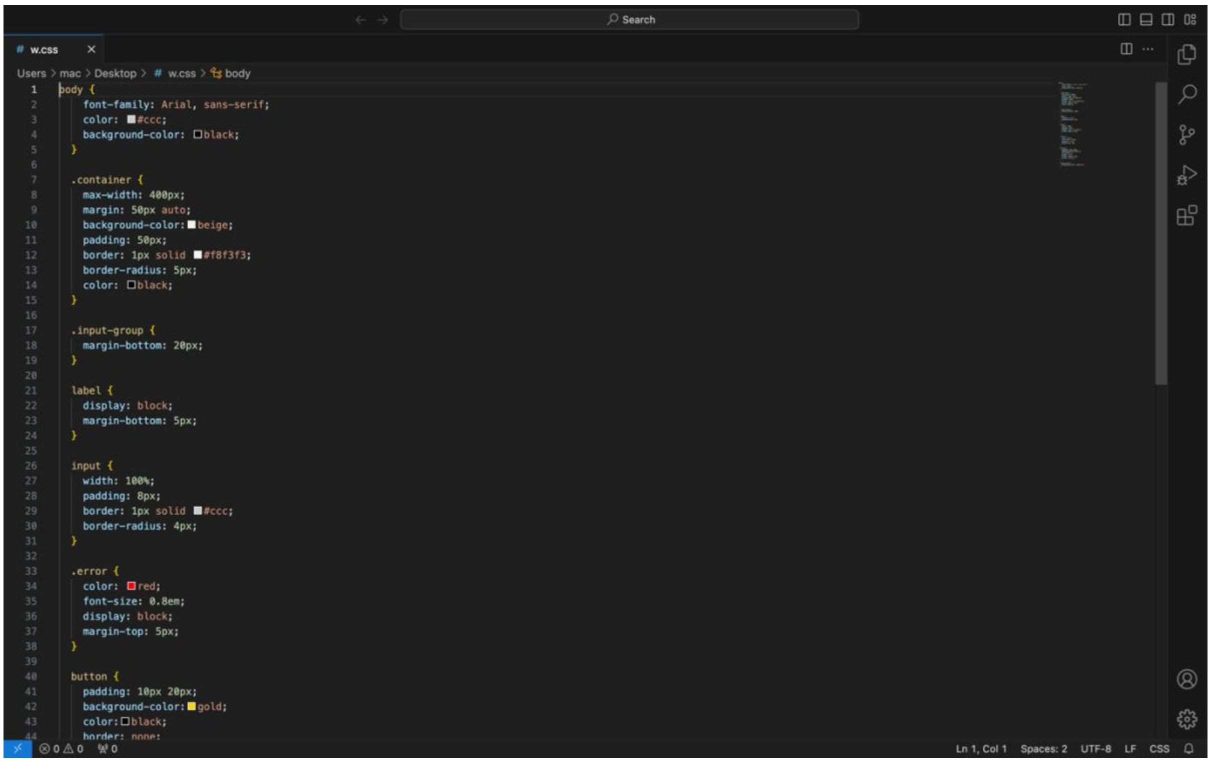
```
Form submitted successfully                                w.js:34
```

OPP-HTML


w.html
Users > mac > Desktop > w.html > html > body > script
Ln 26, Col 19 Spaces: 2 UTF-8 LF HTML

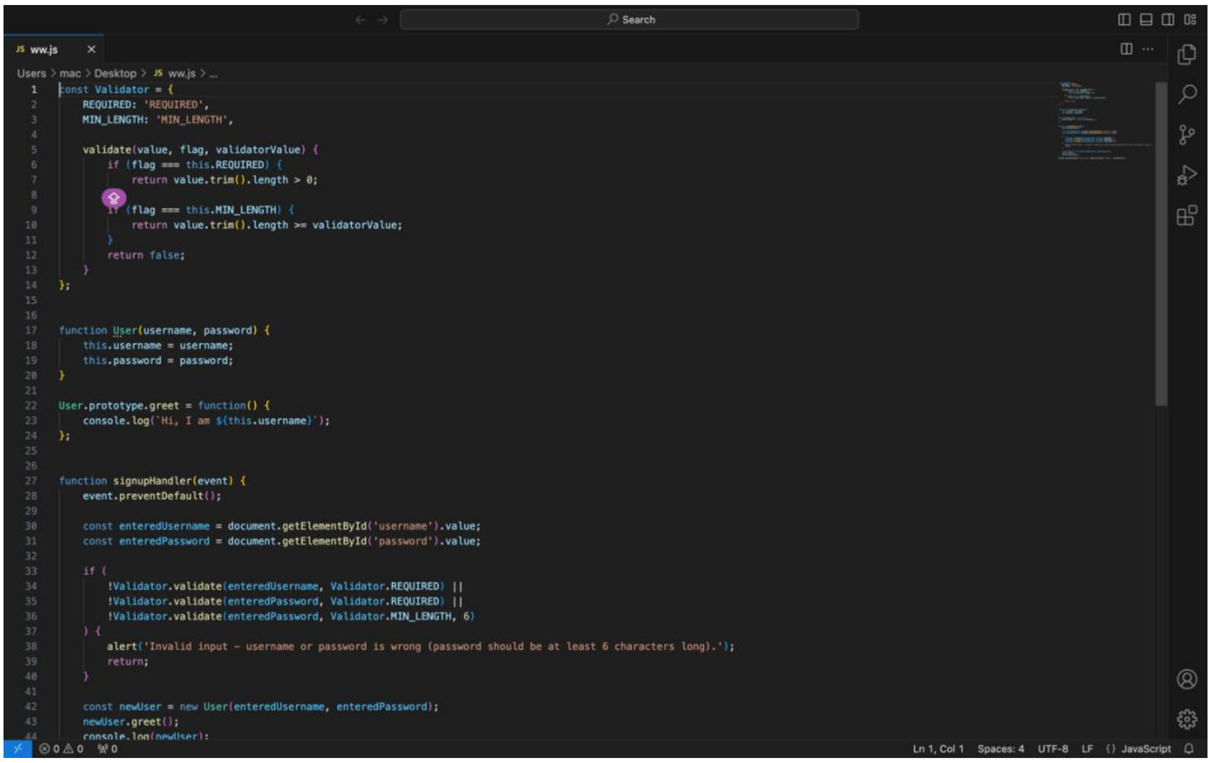
```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login</title>
<link rel="stylesheet" href="w.css">
</head>
<body>
<div class="container">
  <form id="loginForm">
    <h2>Login</h2>
    <div class="input-group">
      <label for="username">Username:</label>
      <input type="text" id="username" name="username">
      <span class="error" id="usernameError"></span>
    </div>
    <div class="input-group">
      <label for="password">Password:</label>
      <input type="password" id="password" name="password">
      <span class="error" id="passwordError"></span>
    </div>
    <button type="button" onclick="validateLoginForm()">Create account</button>
  </form>
</div>
<script src="ww.js"></script>
</body>
</html>
```

OPP-CSS


w.css
Users > mac > Desktop > w.css > body
Ln 1, Col 1 Spaces: 2 UTF-8 LF CSS

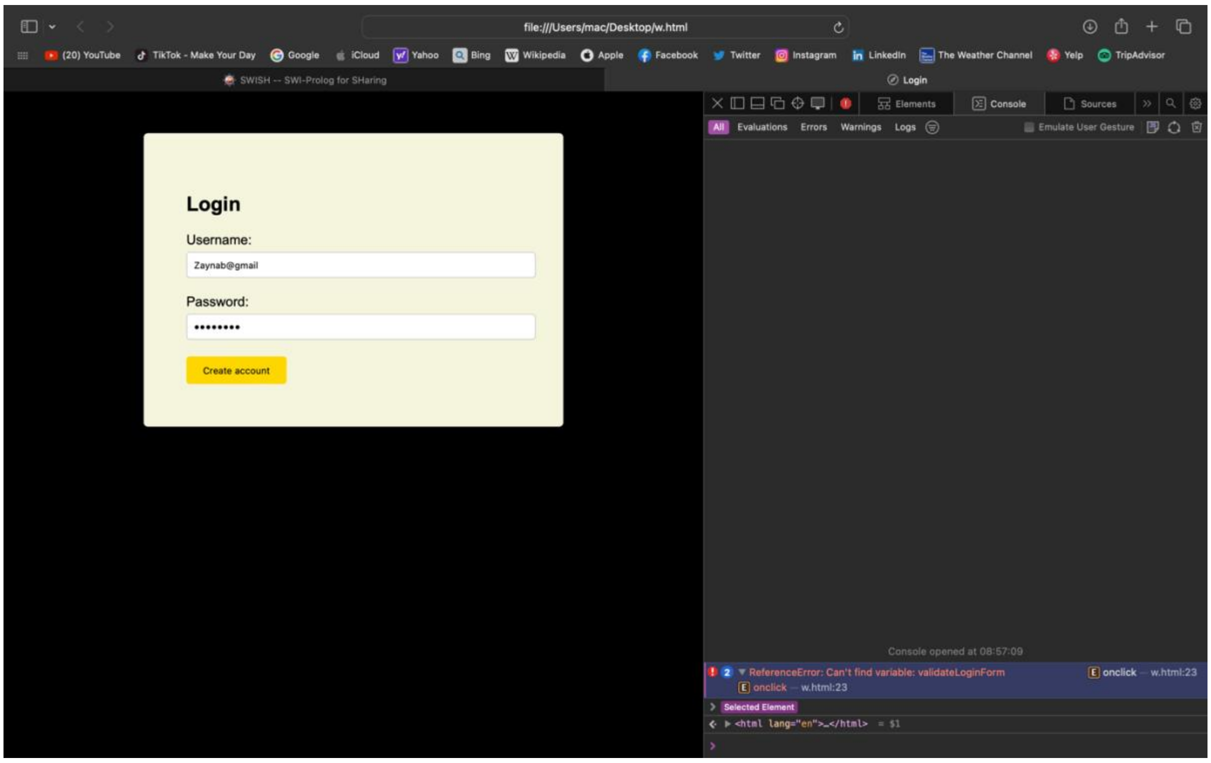
```css
body {
    font-family: Arial, sans-serif;
    color: #ccc;
    background-color: black;
  }

  .container {
    max-width: 400px;
    margin: 50px auto;
    background-color: beige;
    padding: 50px;
    border: 1px solid #f8f3f3;
    border-radius: 5px;
    color: black;
  }

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

  label {
    display: block;
    margin-bottom: 5px;
  }

  input {
    width: 100%;
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 4px;
  }

  .error {
    color: red;
    font-size: 0.8em;
    display: block;
    margin-top: 5px;
  }

  button {
    padding: 10px 20px;
    background-color: gold;
    color: black;
    border: none;
```

OPP-JS



```
const Validator = {
    REQUIRED: 'REQUIRED',
    MIN_LENGTH: 'MIN_LENGTH',

    validate(value, flag, validatorValue) {
        if (flag === this.REQUIRED) {
            return value.trim().length > 0;

        if (flag === this.MIN_LENGTH) {
            return value.trim().length >= validatorValue;
        }
        return false;
    }
};


function User(username, password) {
    this.username = username;
    this.password = password;
}

User.prototype.greet = function() {
    console.log(`Hi, I am ${this.username}`);
};


function signupHandler(event) {
    event.preventDefault();

    const enteredUsername = document.getElementById('username').value;
    const enteredPassword = document.getElementById('password').value;

    if (
        !Validator.validate(enteredUsername, Validator.REQUIRED) ||
        !Validator.validate(enteredPassword, Validator.REQUIRED) ||
        !Validator.validate(enteredPassword, Validator.MIN_LENGTH, 6)
    ) {
        alert('Invalid input - username or password is wrong (password should be at least 6 characters long).');
        return;
    }

    const newUser = new User(enteredUsername, enteredPassword);
    newUser.greet();
    console.log(newUser);
```

OPP-Console



file:///Users/mac/Desktop/w.html

Login

Username: Zaynab@gmail

Password: ••••••••

Create account

```
Console opened at 08:57:09
2 ReferenceError: Can't find variable: validateLoginForm          onclick — w.html:23
    onclick — w.html:23
Selected Element
<html lang="en">…</html> = $1
```