Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования «Тульский государственный университет»

Институт прикладной математики и компьютерных наук

Кафедра «Информационная безопасность»

Отчет по лабораторной работе 8

*по курсу*

СЕТЕВОЕ ПРОГРАММИРОВАНИЕ

Выполнил: студент группы 221061-ПБ Савельевой В. А.\_\_\_\_\_\_\_\_\_\_\_

Проверил: асс.каф.ИБ Поляничко К.В.\_\_\_\_\_\_\_\_\_\_\_

Тула, 2019 г.

**ХОД РАБОТЫ**

1. Сделать генерацию корзины динамической: верстка корзины не должна находиться в HTML-структуре. Там должен быть только div, в который будет вставляться корзина, сгенерированная на базе JS:

a. Пустая корзина должна выводить строку «Корзина пуста»;

b. Наполненная должна выводить «В корзине: n товаров на сумму m рублей».

Продолжаем реализовывать модуль корзины:

• Добавлять в объект корзины выбранные товары по клику на кнопке «Купить» без перезагрузки страницы;

• Привязать к событию покупки товара пересчет корзины и обновление ее внешнего вида. Результат работы программы представлен на рисунке 1.

Листинг 1 - Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Dyson</title>

<link rel = "stylesheet" type = "text/css" href = "style.css">

</head>

<body>

<header class="header">

<h1 align="center">Dyson corporation</h1>

</header>

<div class="container">

<h2>Наши товары</h2>

<div class="products">

<div class="product">

<div>

<h3>Пылесос Dyson Cyclone V10 Parquet</h3> <br>

<img  
src="https://cdn.dyson.ru/data/images/products/dyson-vacuums/cordless/v10/163623/378\_420/1\_163623.jpg" />

</div>

<div>

<b>Двигатель с цифровым управлением Dyson V10</b>

<br>

Компактный, легкий и мощный двигатель для пылесоса Dyson.

<br>

<b>Уборка поверхностей от пола до потолка</b>

<br>

Двигатель пылесоса Dyson Cyclone V10™ размещен над рукояткой, поэтому центр тяжести находится близко к руке, что облегчает уборку высоко расположенных поверхностей и труднодоступных мест.

</div>

<div>

<p class="p11" data-price="39990">Цена: 39 990  
 руб</p>

<button class="add-to-basket-button">Добавить в  
 корзину</button>

</div>

</div>

<div class="product">

<div>

<h3>Фен Dyson Supersonic™ для волос с подставкой  
 (фуксия)</h3><br>

<img src="https://cdn.dyson.ru/data/images/products/dyson-hair-care/hair-driers/179042/378\_420/1\_179042.jpg" />

</div>

<div>

<b>3 точные настройки скорости</b>

<br>

Быстрое высушивание волос, стандартный режим и укладка

<br>

<b>4 уровня нагрева</b>

<br>

100°C - быстрое высушивание волос и укладка<br>

80°C - стандартное высушивание<br>

60°C - бережное высушивание<br>

28°C - поток прохладного воздуха

</div>

<div>

<p class="p11" data-price="34990">Цена: 34 990  
 руб</p>

<button class="add-to-basket-button">Добавить в  
 корзину</button>

</div>

</div>

<div class="product">

<div>

<h3>Персональный очиститель воздуха Dyson Pure Cool Me™ (черный/никель)</h3><br>

<img src="https://cdn.dyson.ru/data/images/products/dyson-fans-and-heaters/purifiers/coolme/164267/378\_420/1\_164267.jpg" />

</div>

<div>

<b>Охлаждает вас</b>

<br>

Очищенный воздух, который дарит прохладу там, где вам необходимо.

<b>Таймер сна</b>

<br>

Вы можете установить таймер сна, выбрав один из интервалов в диапазоне от 30 минут до 8 часов, чтобы получать прохладу и очищенный воздух во время сна и отдыха.

</div>

<div>

<p class="p11" data-price="27990">Цена: 27 990  
 руб</p>

<button class="add-to-basket-button">Добавить в  
 корзину</button>

</div>

</div>

</div>

</div>

<!-- выделяем место под контейнер -->

<div id="basket"></div>

<script src = 'scriptshop.js'></script>

</body>

</html>

Листинг 2 – scriptshop. Js

const basketElement = document.getElementById('basket')

function renderBasket(basek) {

const totalPrice = basket.countTotalPrice()

const totalCount = basket.countTotalCount()

return `

<h1>Ваша корзина</h1>

<p>Кол-во товаров: ${totalCount}</p>

<p>Общая цена: ${totalPrice}</p>

<h2>Товары:</h2>

${basket.products.map(product => {

return `<p>Товар: ${product.name}, кол-во: ${product.count}, цена: ${product.price \* product.count}</p>`

}).join('<br>')}

`

}

class Basket {

constructor(){

this.products = []

}

addToBasket(productToAdd) {

if (this.products.find(product => product.name === productToAdd.name)) {

this.products.find(product => product.name === productToAdd.name).count += productToAdd.count

} else {

this.products.push(productToAdd)

}

this.renderToElem()

}

countTotalPrice() {

let totalPrice = 0

for(let i = 0; i < this.products.length; i++) {

let product = this.products[i]

totalPrice += product.price \* product.count

}

return totalPrice

}

countTotalCount() {

let totalCount = 0

for(let i = 0; i < this.products.length; i++) {

let product = this.products[i]

totalCount += product.count

}

return totalCount

}

renderToElem() {

basketElement.innerHTML = renderBasket(this)

}

}

const basket = new Basket()

basket.renderToElem()

Array.from(document.querySelectorAll('.add-to-basket-button')).forEach(button => {

button.onclick = () => {

const product = button.parentElement.parentElement

const price = +product.querySelector("p").getAttribute("data-price")

const name = product.querySelector("h3").innerText

basket.addToBasket({

price,

count: 1,

name,

})

}

})

Листинг 3 – style.css

h3{

text-align: center;

}

h1{

text-shadow: 2px 2px 20px #000;

}

h2{

text-shadow: 2px 2px 6px #010a1bf5;

font-size:30px;

}

p.p11 {

text-align: center;

font-weight: bold;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

color: #212121;

font-family: sans-serif;

background-color: #FAFAFA;

}

.container {

padding: 8px;

width: 100%;

max-width: 1000px;

margin: 0 auto;

min-height: 800px;

}

#basket {

position: fixed;

bottom: 0;

left: 0;

background-color: #010a1bf5;

width: 350px;

min-height: 100px;

color: white;

padding: 8px;

}

.header {

padding: 40px;

background-color: #093b96;

color: white;

}

.products {

display: flex;

flex-flow: row nowrap;

}

.product {

display: flex;

flex-flow: column nowrap;

justify-content: space-between;

width: 30%;

padding: 8px;

height: 650px;

background-color: white;

margin-right: 15px;

box-shadow: 0 1px 3px rgba(0,0,0,0.12);

}

.product img {

width: 100%;

}

.add-to-basket-button {

width: 100%;

color: white;

padding: 10px;

background-color: #093b96;

//border-radius: 8px;

border: none;

margin-top: 5px;

text-transform: uppercase;

box-shadow: 0 1px 3px rgba(0,0,1,1.82);

cursor: pointer;

}

.add-to-basket-button:hover {

background-color: #8E24AA;

}

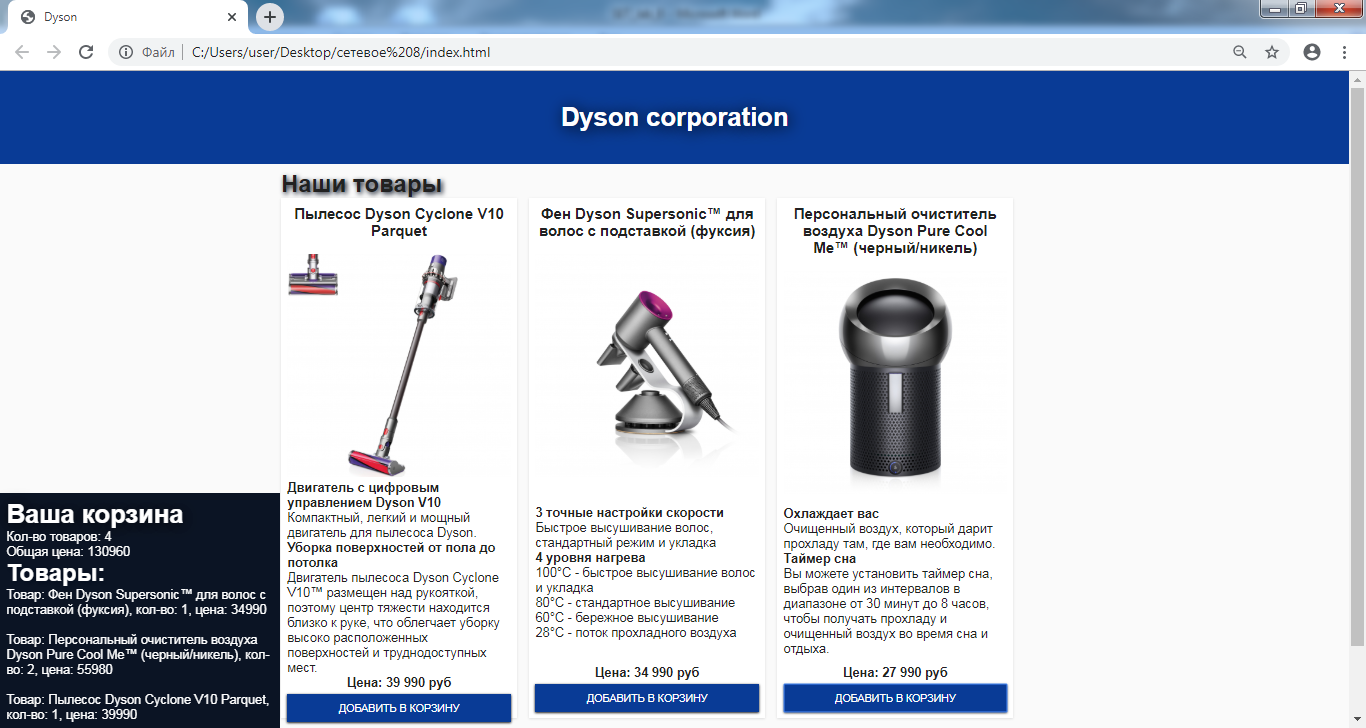


Рисунок 1 – Пример добавления четырех товаров в корзину

1. Создать форму в html со следующими полями:

• Логин - текстовое поле

• Пароль - поле типа password

• Повтор пароля - поле типа password

• Кнопка отправить форму

Необходимо реализовать валидацию этой формы по следующим правилам:

• Логин - должно содержать как минимум 1 символ, не более 50 символов.

• Пароль - минимум 5 символов, максимум 50.

• Повтор пароля - значение должно совпадать с полем пароль.

•Кнопка отправить форму - при нажатии на кнопку форма должна провериться, припрохождении проверки, форма отправляется, если проверка не была пройдена, то:

* + - Каждое поле, которое не прошло проверку должно выделяться красным цветом.
    - Под каждым полем, что не прошло проверку, должна писаться подсказка по какой причине проверка провалилась.

Пользоваться аттрибутами HTML5 запрещено, необходимо реализовать проверки с помощью javascript. Можно пользоваться сторонними стилевыми библиотеками(bulma, bootstrap). Результат тестирования программы представлен на рисунках 2-4.

Листинг 4 - Index.html

<metacharset='utf-8'>

<html>

Логин

<input type = "text" id="login">

Пароль

<input type = "password" id="password">

Повторпароля

<input type = "password" id="password-confirm">

<button>Отправить</button>

<div id="errors" style="color: black"></div>

</html>

<script src = 'scriptPASS.js'></script>

Листинг 5 - scriptPASS.js

const loginInput = document.getElementById('login')

const passwordInput = document.getElementById('password')

const passwordConfirmInput = document.getElementById('password-confirm')

const errorsElement = document.getElementById('errors')

loginInput.addEventListener('input', validate)// обработчик событий

passwordInput.addEventListener('input', validate)

passwordConfirmInput.addEventListener('input', validate)

loginInput.onchange = validate//событие обработки ошибки при изменении содержимого

passwordInput.onchange = validate

passwordConfirmInput.onchange = validate

function getErrors() {

if (loginInput.value.length === 0) {

loginInput.style.borderColor = 'red'

return 'Не заполнено поле login'

}

if (loginInput.value.length > 50) {

loginInput.style.borderColor = 'red'

return 'Логин не должен превышать 50 символов'

}

if (passwordInput.value.length < 5) {

passwordInput.style.borderColor = 'red'

return 'Пароль не может быть меньше 5 символов'

}

if (passwordInput.valuelength > 50) {

passwordInput.style.borderColor = 'red'

return 'Пароль не должен превышать 50 символов'

}

if(passwordInput.value !== passwordConfirmInput.value) {

passwordInput.style.borderColor = 'red'

passwordConfirmInput.style.borderColor = 'red'

return 'Поля "пароль" и "подтверждение пароля" не совпадают'

}

loginInput.style.borderColor = ''

passwordInput.style.borderColor = ''

passwordConfirmInput.style.borderColor = ''

return false

}

function validate() {

const errors = getErrors()

if (errors) {

errorsElement.innerText = errors

return

}

errorsElement.innerText = ''//обнуляем для след запроса

}

Листинг 6 - style.css

table {

font-family: Impact, sans-serif;

font-size: 30px;

background-color: white;

color: black;

}

td {

height: 50px;

width: 50px;

text-align: center;

vertical-align: middle;

}

/\* все четные номера строк, нечетные номера столбцов\*/

tr:nth-child(even) td:nth-child(odd) {

background-color: black;

}

tr:nth-child(odd) td:nth-child(even) {

background-color: black;

}

/\*последний элемент родителя, нечетные номера столбцов \*/

tr:last-childtd:nth-child(odd),

/\*первый элемент родителя, четные номера столбцов \*/

tr:first-childtd:nth-child(even){

background-color: white;

}

tr:nth-child(even) td:first-child{

background-color: white;

}

tr:nth-child(odd) td:last-child{

background-color: white;

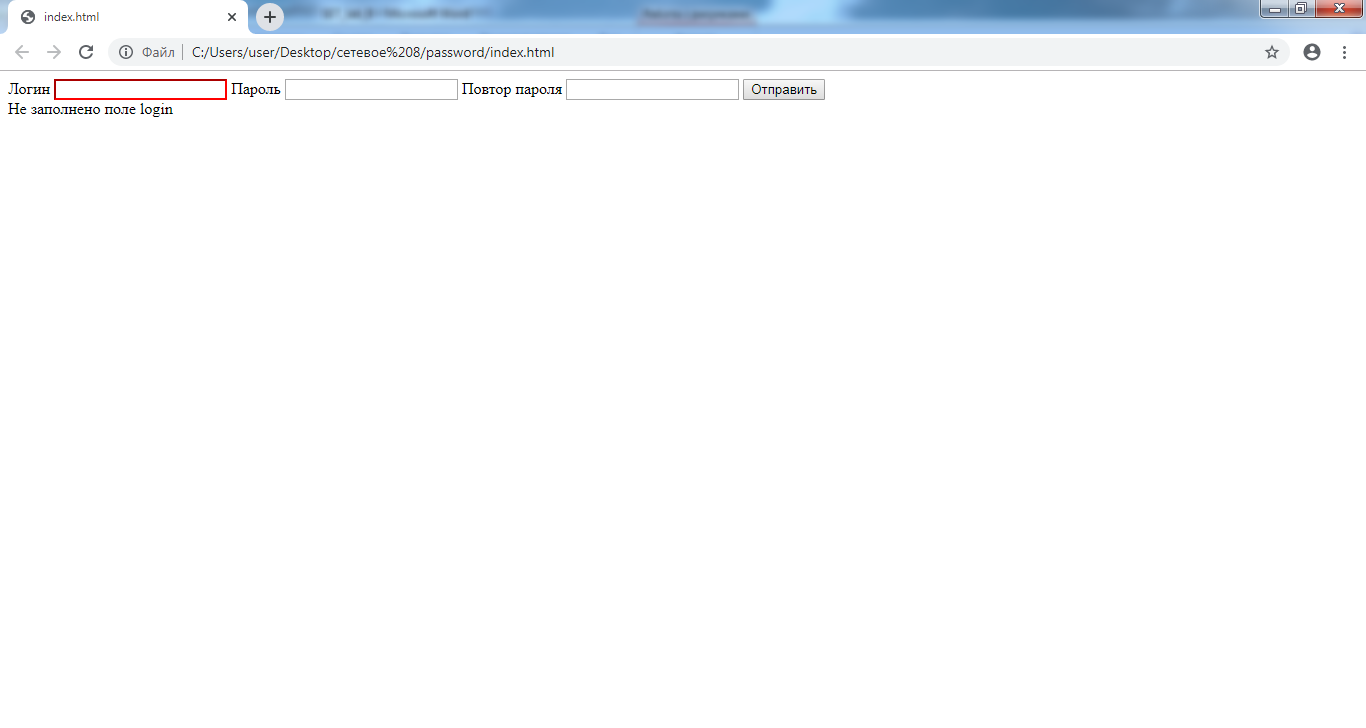


Рисунок 2 – Вывод ошибки не заполненного поля login

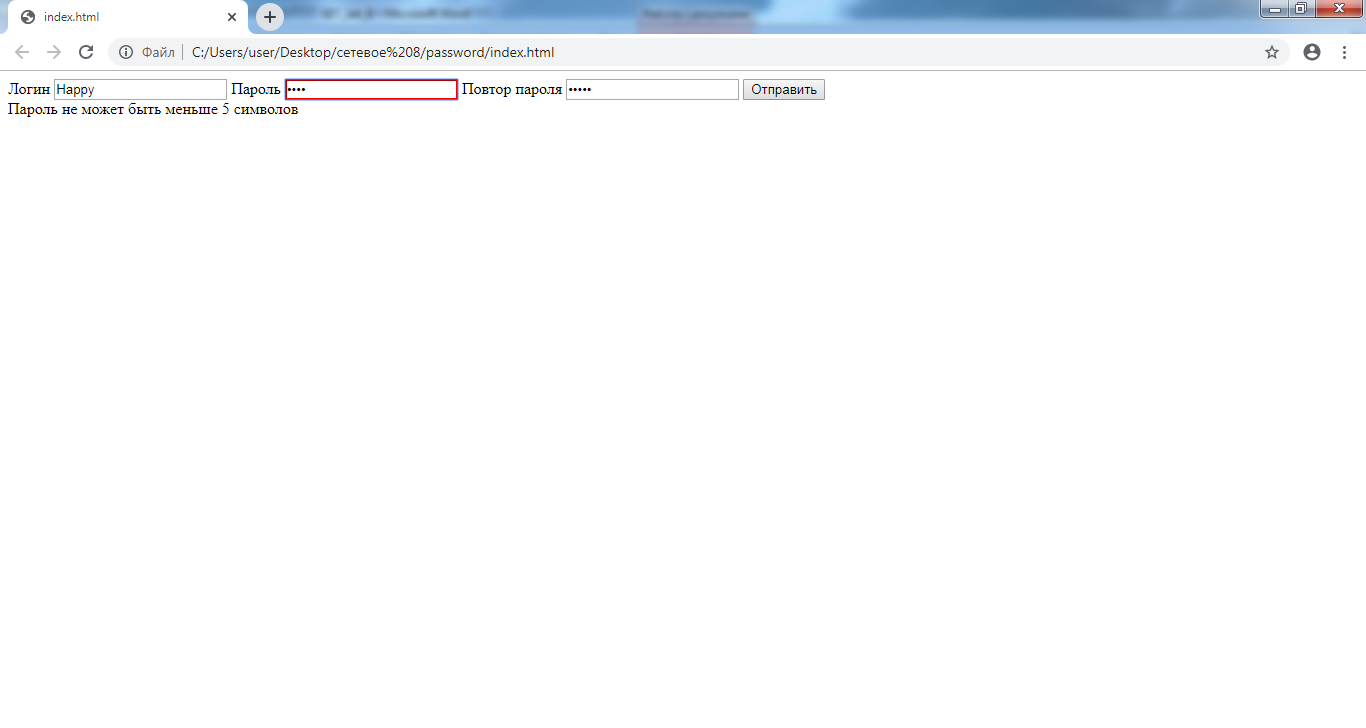


Рисунок 3 – Вывод ошибки короткого пароля

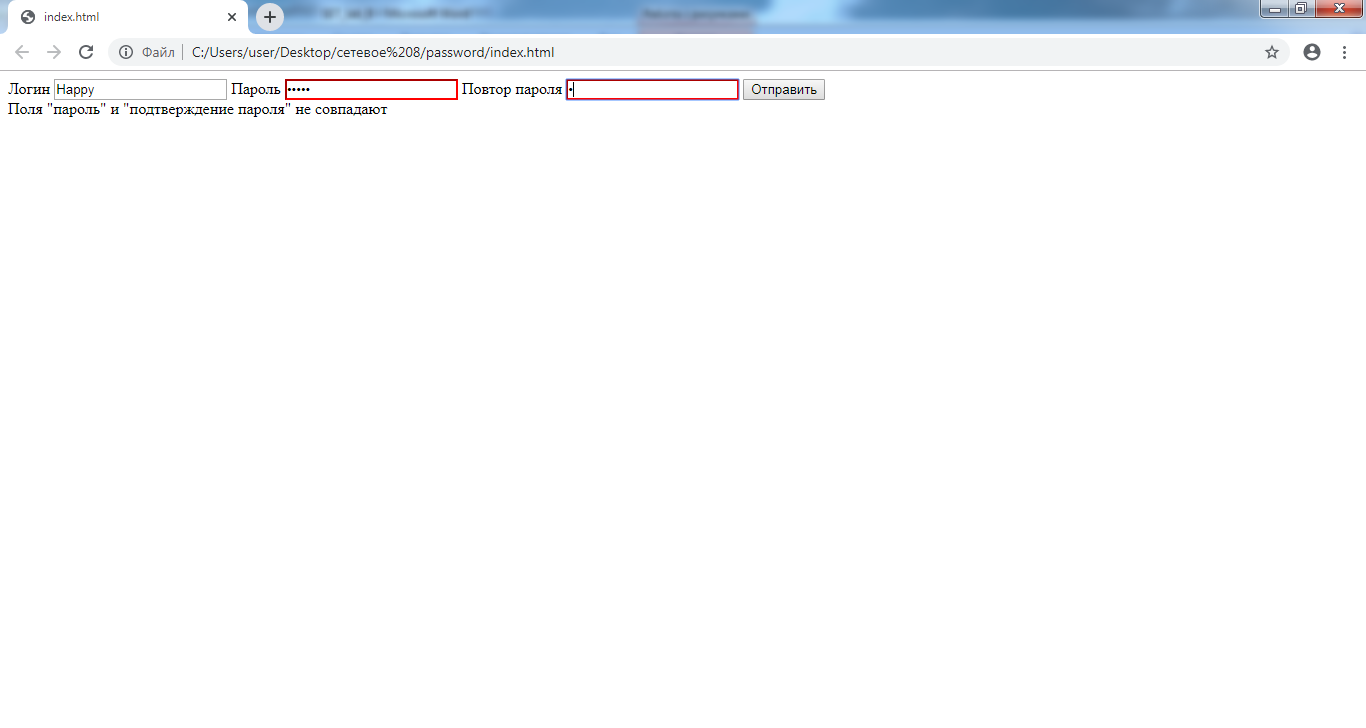


Рисунок 4 – Вывод ошибки разного пароля

# ВЫВОД

# При выполнении данной лабораторной работы были изучены навыки программирования на языке JavaScript, а так же понятие браузерного события и написаны демонстрационные программы для закрепления материала.