|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.03 «Прикладная информатика»**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 4 |

**Название:**

Javascript. Создание сложных HTML-страниц.

Формирование таблиц

**Дисциплина:** Языки Интернет-программирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-34Б |  | 30.09.2022 | С. А. Рахманов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | Д. В. Малахов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Цель работы:** закрепление навыков программирования на языке Javascript с использованием модели документа DOM, навыков управления реакцией элементов на страницах HTML и отладки Javascript-приложений.

**Задание:**

Разработать страницу для оформления заказа товаров.

1. Реализовать форму для ввода наименования и стоимости. При нажатии на кнопку «Добавить», добавить введенные значения в таблицу.
2. Сделать кнопку расчёта стоимости заказа.
3. Поставить обработчик на изменение фона ячеек таблицы со стоимостью в тот момент, когда над ними находится указатель мыши.
4. Подготовить страницу, содержащую таблицу товаров и их стоимости. На базе этой страницы обеспечить возможность выбора строк и добавления их в таблицу формирования заказа.

Примечание: в качестве усложненного варианта задания допустимо использовать библиотеки jQuery и Dojo.

**Код страниц с программами:**

**l4.1.html**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Lab work 4.1</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"

crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"

integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"

crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>

<script src="script4.1.js"></script>

<style>

.price {

position: fixed;

bottom: 1rem;

left: 50%;

transform: translate(-50%, 0);

}

.main {

margin-bottom: 100px;

/\* background-color:rgba(0, 98, 255, 0.551); \*/

}

</style>

</head>

<body>

<div class="main">

<table class="table table-hover" id="product-table">

<thead>

<tr>

<th scope="col">№</th>

<th scope="col">Наименование</th>

<th scope="col">Стоимость</th>

</tr>

</thead>

<tbody>

</tbody>

<tfoot>

</tfoot>

</table>

<div style="padding: 10px;">

<form class="row g-3">

<div class="col-sm-8 col-md-6 col-lg-4">

<div class="">

<label class="form-label"></label>

<input type="text" class="form-control" id="nameProductForm" placeholder="Наименование товара">

</div>

<div class="mt-3">

<input type="number" class="form-control" id="price-text" value="1000">

<input type="range" class="form-range mt-4" min="0" max="10000" value="1000" id="price-range" oninput="changeData()">

</div>

</div>

</form>

<div class="col-12">

<button class="btn btn-primary mt-4" type="submit" onclick="addProduct()">Добавить</button>

</div>

</div>

</div>

<div class="price">

<div class="row align-items-end">

<div class="col">

<button type="button" class="btn btn-primary btn-lg" onclick="evaluateCost()">Рассичитать стоимость</button>

</div>

</div>

</div>

</body>

</html>

**l4.2.html**

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="UTF-8">

<title>Lab work 4.2</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet"

integrity="sha384-iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js"

integrity="sha384-u1OknCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"

crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.1/jquery.min.js"></script>

<script src="script4.2.js"></script>

<style>

.price {

position: fixed;

bottom: 1rem;

left: 50%;

/\*transform: translate(-50%, 0);\*/

}

.main {

margin-bottom: 100px;

}

.input-block {

display: inline-block;

}

#calc-res {

display: inline-block;

margin-left: 2rem;

font-weight: bold;

font-size: larger;

color: rgba(1, 73, 255, 0.766);

}

.text1 {

text-align: center;

transform: translate(-8%, 0);

}

.text2 {

text-align: left;

}

</style>

</head>

<body>

<div class="row">

<div class="col-9">

<h1 class="text1">Товары</h1>

</div>

<div class="col-3">

<h1 class="text2">Заказ</h1>

</div>

<div class="col-md-8">

<table class=" table table-hover">

<thead>

<tr>

<th scope="col">№</th>

<th scope="col">Наименование</th>

<th scope="col">Стоимость</th>

</tr>

</thead>

<tbody id="product-table">

<tr>

<th scope="row">1</th>

<td>Брелок</td>

<td>1000</td>

</tr>

<tr>

<th scope="row">2</th>

<td>Носки</td>

<td>2000</td>

</tr>

<tr>

<th scope="row">3</th>

<td>Шапка</td>

<td>3000</td>

</tr>

<tr>

<th scope="row">4</th>

<td>Куртка</td>

<td>4000</td>

</tr>

<tr>

<th scope="row">5</th>

<td>Штаны</td>

<td>5000</td>

</tr>

<tr>

<th scope="row">6</th>

<td>Кофта</td>

<td>6000</td>

</tr>

<tr>

<th scope="row">7</th>

<td>Обувь</td>

<td>7000</td>

</tr>

</tbody>

<tfoot>

</tfoot>

</table>

</div>

<div class="col-md-1"></div>

<div class="col-md-3 col-lg-3">

<table class="table table-sm table-hover">

<thead>

<tr>

<th scope="col">Наименование</th>

<th scope="col">Стоимость</th>

</tr>

</thead>

<tbody id="order-table">

</tbody>

<tfoot>

</tfoot>

</table>

<div class="input-block">

<button type="submit" class="btn btn-primary btm-sm" id="calc">Расчитать стоимость</button>

<div id="calc-res"></div>

</div>

</div>

</div>

</body>

</html>

**script4.1.js**

// меняет данные при движении ползунка

function changeData() {

let data = document.getElementById("price-range").value.toString();

document.getElementById("price-text").value = data;

}

// создает новую строку

function createNewProductRow(tbody, number) {

let newRow = tbody.insertRow();

newRow.setAttribute('id', number);

let numberCell = newRow.insertCell();

let nameCell = newRow.insertCell();

let costCell = newRow.insertCell();

return Array(numberCell, nameCell, costCell);

}

//создает новый набор значений для строки

function createNewTextNodes(number, name, price) {

let numberText = document.createTextNode(number);

let nameText = document.createTextNode(name);

let costText = document.createTextNode(price);

return Array(numberText, nameText, costText);

}

// добавляет набор значений в строку и возвращает её

function addTextNodesToCells(productRow, productRowTexts) {

productRow[0].appendChild(productRowTexts[0]);

productRow[1].appendChild(productRowTexts[1]);

productRow[2].appendChild(productRowTexts[2]);

return productRow;

}

// очистка поля с наименованием

function clearInputField() {

document.getElementById("nameProductForm").value = "";

}

//меняет стиль строки

function changeStyleFontWeight(productRow) {

productRow[0].style = "font-weight: bold";

}

// внесение курсора

function modifyCell(element) {

element.style = "color: white; background-color:rgba(0, 98, 255, 0.551); transition-property: backgroundcolor; transition-duration: 4s; transition-delay: 2s;"

}

// вынесение курсора

function modifyCellToOrdinary(element) {

element.style = "color: black; background-color: white;";

}

function addProduct() {

let name = document.getElementById("nameProductForm").value;

let price = document.getElementById("price-text").value;

let table = document.getElementById('product-table');

let number = table.tBodies[0].rows.length + 1;

try {

price = parseInt(price);

if (name == '' || isNaN(price) || price < 0) {

throw new Error();

}

} catch(err) {

console.log(err);

alert('error');

return 1;

}

//

let tbodyRef = document.getElementById('product-table').getElementsByTagName('tbody')[0];

let newProductRow = createNewProductRow(tbodyRef, number);

newProductRow[2].addEventListener('mouseover', function () { modifyCell(newProductRow[2]); }, true);

newProductRow[2].addEventListener('mouseout', function () { modifyCellToOrdinary(newProductRow[2]); }, true);

let newProductRowTexts = createNewTextNodes(number, name, price);

changeStyleFontWeight(newProductRow);

clearInputField();

addTextNodesToCells(newProductRow, newProductRowTexts);

if (document.getElementById('warning') != null) {

document.getElementById('warning').textContent = "данные о стоимости заказа устарели, пожалуйста - обновите их"

document.getElementById('warning').style = "color: #F03457; font-size: 10px;";

}

}

function setIds(summaryRow) {

summaryRow[1].setAttribute('id', 'warning');

summaryRow[2].setAttribute('id', 'cost-of-all');

return summaryRow;

}

function fillTextNodesSummaryRow(orderCost) {

let summaryText = document.createTextNode("Итого");

let emptyText = document.createTextNode("");

let costText = document.createTextNode(orderCost);

return Array(summaryText, emptyText, costText);

}

function createSummaryRow(orderCost) {

let tfootRef = document.getElementById('product-table').getElementsByTagName('tfoot')[0];

let summaryRow = createNewProductRow(tfootRef, 0);

summaryRow = setIds(summaryRow);

clearInputField();

let summaryTextNodes = fillTextNodesSummaryRow(orderCost);

changeStyleFontWeight(summaryRow);

addTextNodesToCells(summaryRow, summaryTextNodes);

}

function countOrderCost(table) {

let tbodyRowCount = table.tBodies[0].rows.length;

let orderCost = 0;

for (let i = 0; i < tbodyRowCount; i++) {

orderCost += Number(table.tBodies[0].rows[i].cells[2].innerHTML);

}

return orderCost;

}

let flag = true;

function evaluateCost() {

let table = document.getElementById('product-table');

let orderCost = countOrderCost(table);

console.log(orderCost);

if (flag) {

createSummaryRow(orderCost);

flag = false;

} else {

document.getElementById('cost-of-all').textContent = orderCost;

if (document.getElementById('warning').textContent != "") {

document.getElementById('warning').textContent = "";

}

}

}

**script4.2.js**

window.onload = function (e) { addRowEventListeners() };

let priceses = 0;

function addRowEventListeners() {

let rows = document.getElementById('product-table').getElementsByTagName('tr');

let table\_orders = document.getElementById('order-table');

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

rows[i].addEventListener('click', function(event){

let tr = event.target.closest('tr');

let tds = tr.getElementsByTagName('td');

let label = tds[0].innerText;

let price = parseInt(tds[1].innerText);

priceses += price;

let html = `

<tr>

<td>`+label+`</td>

<td>`+price+`</td>

</tr>

`;

table\_orders.innerHTML = html + table\_orders.innerHTML;

});

}

document.getElementById('calc').addEventListener('click', calcTotalPrice);

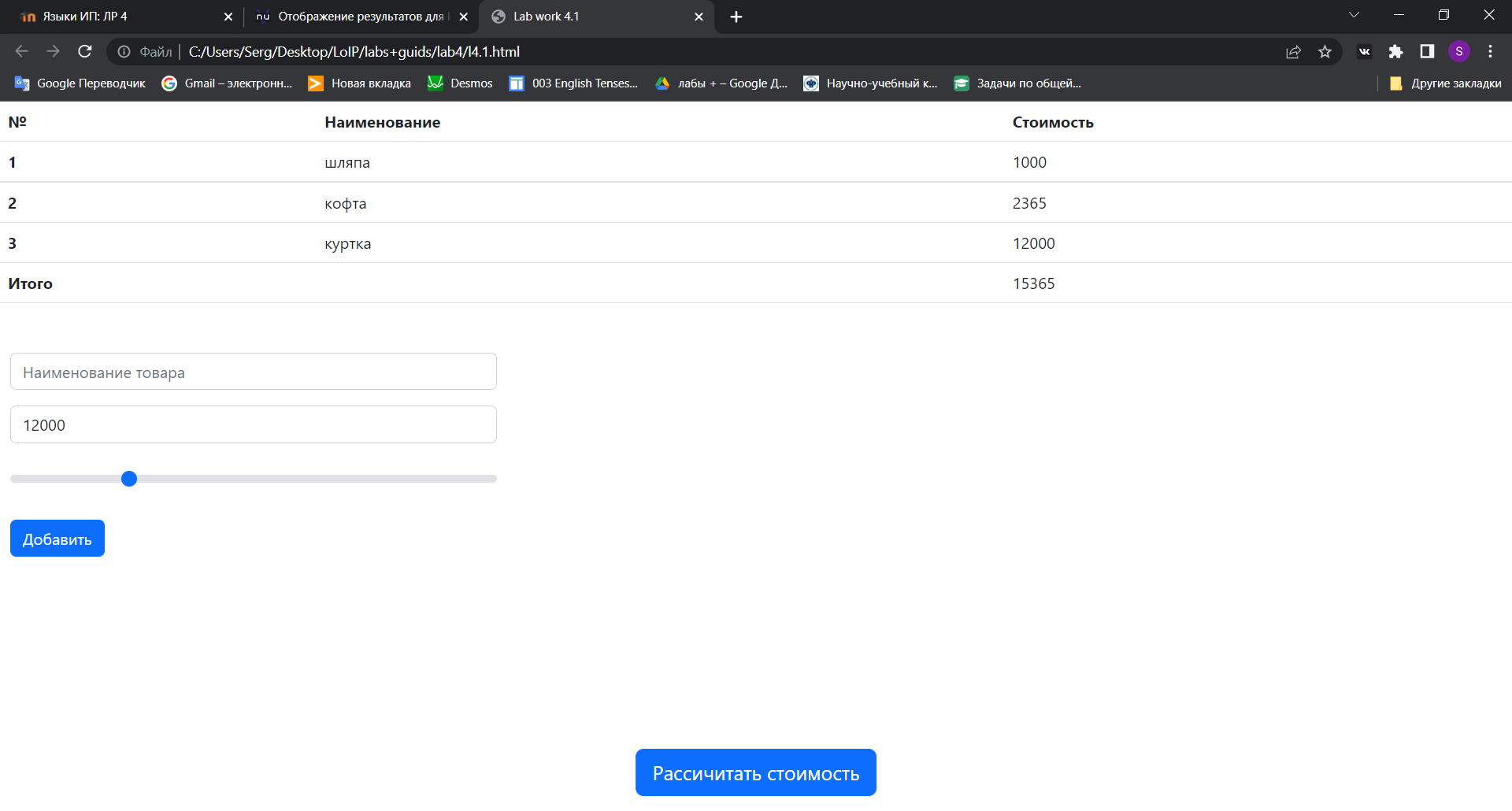
}

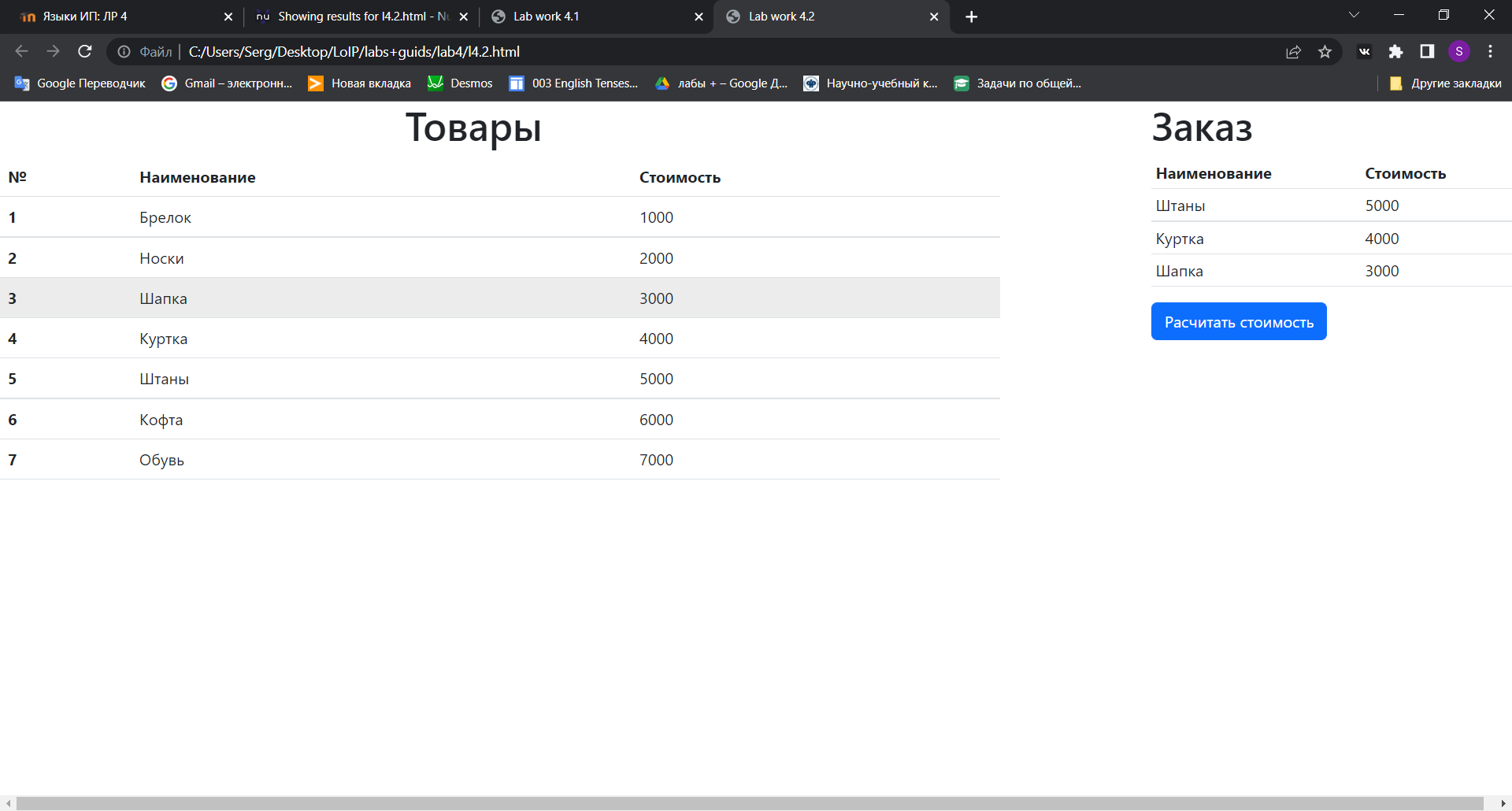
function calcTotalPrice() {

alert('Всего: '+priceses);

}

**Страницы:**

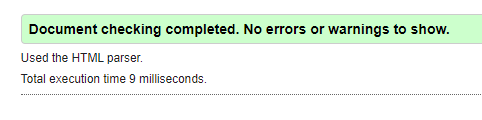




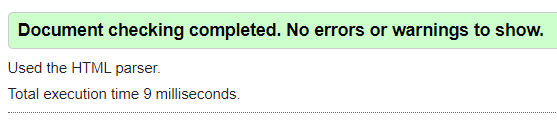
|  |  |
| --- | --- |
| **Ошибки валидации** | **Проявление в браузере** |
|  | **-** |
|  | **-** |
|  | **-** |
|  | **-** |
|  | **-** |

**После устранения ошибок:**

**l4.1.html**



**l4.2.html**



**Вывод:** закрепил навыки программирования на языке Javascript с использованием модели документа DOM, навыки управления реакцией элементов на страницах HTML и отладки Javascript-приложений.