

Лабораторная работа №3

«Исследование объектной модели документа (DOM) и системы событий JavaScript»

Цель работы

Исследовать структуру модели документа DOM. Изучить динамическую объектную модель документа, предоставляемую стандартом DOM и систему событий языка JavaScript, возможность хранения данных на стороне клиента. Приобрести практические навыки работы с событиями JavaScript, деревом документа, Local Storage и Cookies.

Ход работы

1. Реализовать интерактивное графическое меню сайта, в соответствии со следующими требованиями:

- при наведении мыши на соответствующий пункт меню изображение, соответствующее этому пункту меняется на другое. В том случае если этот пункт не активен, т.е. не загружена соответствующая ему страница.



Рисунок 1 – Изменение картинки при наведении и на активной странице сайта

- сделать пункт «Мои интересы» в виде выпадающего меню, каждый элемент которого ведет на соответствующую ему подсекцию данной страницы.

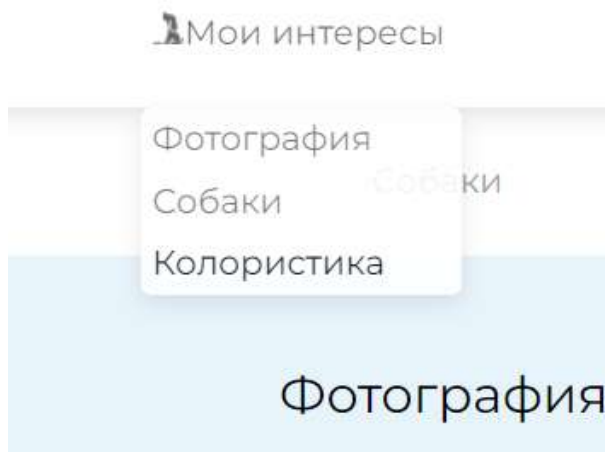


Рисунок 2 – Выпадающее меню раскрывается по наведению

2. Реализовать отображение в области меню сайта текущих даты и времени (обновление времени 1 раз в секунду). Формат даты: ЧЧ Месяц ГГ.



Рисунок 3 – Дата и время в области меню сайта

Код программы:

```
function dropMenu(){
  const elem1 = document.getElementById("categories");
  elem1.onmouseover = function(event){
    if(elem1!==null){
      event.target.nextElementSibling.classList.toggle('sub-menu-active');
      // console.log(event);
    }
  }
}
// // const elem2 = document.getElementById("categories");
// elem1.onmouseleave = function(event){
//   if(elem1){
//     event.target.nextElementSibling.classList.remove('sub-menu-active');
//     // console.log(event);
//   }
// }

// elem1.onmouseleave

// let now = new Date();

// console.log(now.getDate()); //0..31
// console.log(now.getMonth()); //0..11
// console.log(now.getFullYear());
// console.log(now.getHours()); //0..24
// console.log(now.getMinutes()); //0..59

const month = ['Января', 'Февраля', 'Марта', 'Апреля', 'Мая', 'Июня', 'Июля', 'Августа', 'Сентября', 'Ноября', 'Декабря']

function addLeavingZero(d){
```

```

        return (d < 10) ? '0' + d : d;
    }

function getUserTime (){

    var d = document.createElement("div");

    d.style.fontWeight='100';
    d.style.fontSize='13px';
    d.style.textAlign='center';

    var z = document.createElement('p');

    d.appendChild(z);
    document.getElementById('logo').appendChild(d);

    let t = new Date();
    let D = addLeavingZero(t.getDate());
    let M = month[t.getMonth()-1];
    let Y = t.getFullYear() % 100;

    let h = addLeavingZero(t.getHours());
    let m = addLeavingZero (t.getMinutes());

    z.innerHTML = `${D} ${M} ${Y} ${h}:${m}`;

    setInterval(function(){
        let t = new Date();
        let D = addLeavingZero(t.getDate());
        let M = month[t.getMonth()-1];
        let Y = t.getFullYear() % 100;

        let h = addLeavingZero(t.getHours());
        let m = addLeavingZero (t.getMinutes());

        z.innerHTML = `${D} ${M} ${Y} ${h}:${m}`;

    }, 1000)

}

//ЧЧ Месяц ГГ

function menuPicture() {
    var choiceMenu = document.getElementsByClassName("forPicture");

    for (var i=0; i<choiceMenu.length; i++){

        var leafs = choiceMenu[i].childNodes;
        leafs[0].src = "./img/firstdogfotmenu.png";

        var location = window.location.href; //текущая страница
        var link = choiceMenu[i].href;

        if(link == location) {
            var leafs = choiceMenu[i].childNodes;
            leafs[0].src = "./img/seconddogformenu.png";
            continue;
        }

        choiceMenu[i].addEventListener("mouseover", function(e){
            var leafs = this.childNodes;
            leafs[0].src = "./img/seconddogformenu.png";
        })

        choiceMenu[i].addEventListener("mouseout", function(e){

```

```

        var leafs = this.childNodes;
        leafs[0].src = "./img/firstdogfotmenu.png";
    })

};

// for (var i=0; i<choiceMenu.length; i++){

//     var location = window.location.href; //текущая страница

//     var link = choiceMenu[i].href;

//     if(link == location) {
//         var leafs = choiceMenu[i].childNodes;
//         leafs[0].src = "./img/seconddogformenu.png";
//     }
// };
}

getUserTime();
menuPicture();
dropMenu();

```

3. На странице «Контакт» добавить поле «Дата рождения», для которого реализовать всплывающий снизу элемент «календарь».

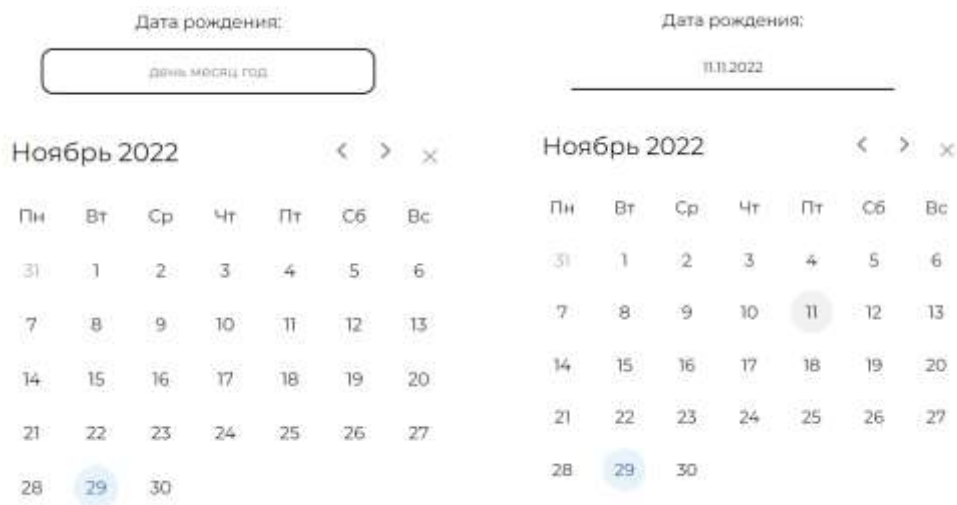


Рисунок 4 – (а) Отображение текущей даты в календаре, (б) Выбор любой даты и запись ее в поле

Код программы:

```

// const currentDate = document.querySelector(".current-date");
// const daysTag = document.querySelector(".days");
// weekTag = document.querySelector(".weeks");
// iconChevron = document.querySelector(".icons");
// calendar = document.querySelector(".calendar");
wrapper = document.querySelector(".wrapper");
// header = document.querySelector("header");

//shooba
var openCalendarFlag = false;
var header = document.createElement("header");

```

```

var currentDate = document.createElement("p");
currentDate.classList.add("current-date");

var iconChevron = document.createElement("div");
iconChevron.classList.add("icons");

header.appendChild(currentDate);
header.appendChild(iconChevron);

var calendar = document.createElement("div");
calendar.classList.add("calendar");

var weekTag = document.createElement("ul");
var daysTag = document.createElement("ul");
weekTag.classList.add("weeks");
daysTag.classList.add("days");

calendar.appendChild(weekTag);
calendar.appendChild(daysTag);
//booba

let date = new Date(),
currYear = date.getFullYear(),
currMonth = date.getMonth();

const months =
['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль', 'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь'];
const weekday = ["Пн", "Вт", "Ср", "Чт", "Пт", "Сб", "Вс"];

function renderCalendar() {
    var weekLi;
    let firstDayOfMonth = new Date (currYear, currMonth, 0).getDay(), // порядковый номер дня недели указанной
    даты по местному времени, где 0 соответствует воскресенью
    lastDayofLastMonth = new Date (currYear, currMonth, 0).getDate(),
    lastDateofMonth = new Date (currYear, currMonth + 1, 0).getDate();//последнее число месяца
    let liTag = "";

    for (let i=firstDayOfMonth; i>0; i--){
        liTag += `<li class="inactive">${lastDayofLastMonth - i + 1}</li>`;
    }

    for (let i=1; i <= lastDateofMonth; i++){
        let isToday = i === date.getDate() && currMonth === new Date().getMonth() && currYear === new
Date().getFullYear() ? "active" : "otherDays" ;
        liTag += `<li class="${isToday}">${i}</li>`;
    }

    calendar.appendChild(daysTag);
    daysTag.innerHTML = liTag;

    let daysInThisMonths = daysTag.getElementsByClassName("otherDays");
    for (let i=0; i<daysInThisMonths.length; i++){
        daysInThisMonths[i].addEventListener("click", function(e){
            document.getElementById("bday").value = this.innerHTML + "." + (Number(currMonth) + Number(1)) + "." +
+ currYear;
        });
    }

    currentDate.innerHTML = `${months[currMonth]} ${currYear}`;

    let iconLeft = document.createElement("span");
    iconLeft.innerText= "chevron_left";
    iconLeft.className = "material-symbols-rounded";
    iconLeft.id = "prev";
    iconChevron.appendChild(iconLeft);

    iconLeft.addEventListener("click", () => {
        currMonth = (iconLeft.id === "prev") ? currMonth - 1 : currMonth +1;
    });
}

```

```

    if(currMonth < 0 || currMonth > 11){
        date = new Date(currYear, currMonth);
        currYear = date.getFullYear();
        currMonth = date.getMonth();
    } else {
        date = new Date();
    }

    document.getElementsByClassName("weeks")[0].innerHTML="";
    document.getElementsByClassName("icons")[0].innerHTML="";

    // удаление дней недели
    let del = weekTag.childNodes.length;
    for(var i=del-1; i>=0; i--){
        weekTag.childNodes[i].remove();
    }

    renderCalendar();
});

let iconRight = document.createElement("span");
iconRight.innerText= "chevron_right";
iconRight.className = "material-symbols-rounded";
iconRight.id = "next";
iconChevron.appendChild(iconRight);

iconRight.addEventListener("click", () => {
    currMonth = (iconRight.id === "prev") ? currMonth - 1 : currMonth +1;

    if(currMonth < 0 || currMonth > 11){
        date = new Date(currYear, currMonth);
        currYear = date.getFullYear();
        currMonth = date.getMonth();
    } else {
        date = new Date();
    }

    document.getElementsByClassName("weeks")[0].innerHTML="";
    document.getElementsByClassName("icons")[0].innerHTML="";

    // удаление дней недели
    let del = weekTag.childNodes.length;
    for(var i=del-1; i>=0; i--){
        weekTag.childNodes[i].remove();
    }

    renderCalendar();
});

// добавление крестика
let iconCross = document.createElement("span");
iconCross.innerHTML= "&times";
iconCross.className = "material-symbols-rounded";
iconCross.style.alignItems = "center";
iconChevron.appendChild(iconCross);

// закрытие (удаление) при нажатии на крестик
iconCross.addEventListener("click", function(){
    let del = wrapper.childNodes.length;
    let delHeader = iconChevron.childNodes.length;
    let delByCross = weekTag.childNodes.length;
    openCalendarFlag = false;

    for(var i=delHeader-1; i>=0; i--){
        console.log(header.childNodes);
        iconChevron.childNodes[i].remove();
    }

```

```

        for(var i=delByCross-1; i>=0; i--){
            weekTag.childNodes[i].remove();
        }

        for(var i=del-1; i>=0; i--){
            wrapper.childNodes[i].remove();
        }
    });

    for (let i=0; i<7; i++){
        weekLi = document.createElement("li");
        weekLi.innerText = `${weekday[i]}`;
        weekTag.appendChild(weekLi);
    }

    header.appendChild(currentDate);
    header.appendChild(iconChevron);

    calendar.appendChild(weekTag);
    calendar.appendChild(daysTag);

    wrapper.appendChild(header);
    wrapper.appendChild(calendar);
}

// prevNextIcon = document.querySelectorAll(".icons span");

// prevNextIcon.forEach(element => {
//     element.addEventListener("click", () => {
//         currMonth = (element.id === "prev") ? currMonth - 1 : currMonth +1;

//         if(currMonth < 0 || currMonth > 11){
//             date = new Date(currYear, currMonth);
//             currYear = date.getFullYear();
//             currMonth = date.getMonth();
//         } else {
//             date = new Date();
//         }

//         document.getElementsByClassName("weeks")[0].innerHTML="";
//         document.getElementsByClassName("icons")[0].innerHTML="";

//         renderCalendar();
//     });
// });

function help(){
    var tt = document.getElementById("bday");
    tt.addEventListener("focus", function(e){
        if (!openCalendarFlag){
            renderCalendar();
            openCalendarFlag = true;
        }
    });
    // tt.addEventListener("blur", del);

    function del(){
        var del = wrapper.childNodes.length;

        for(var i=del-1; i>=0; i--){
            wrapper.childNodes[i].remove();
        }
    }
}

```

```

help();

// console.log(wrapper.childNodes.length);

// установка выбранного месяца в строку с датой
// function setMonth(){
//     var monthInp = Number(currMonth) + Number(1);
//     if (monthInp < 10) monthInp = "0" + Number(monthInp);
//     var inp = document.getElementById("bday");
//     inp.value = "01.01.2022";
//     inp.value = inp.value.substring(0, 3) + monthInp + inp.value.substring(5);
// }

// установка выбранного года в строку с датой
// function setDay(nday){
//     var dayInp = Number(nday);
//     if (dayInp < 10) dayInp = "0" + Number(dayInp);
//     inp.value = dayInp + inp.value.substring(2);
// }

```

4. Реализовать динамическую проверку корректности заполнения пользователем формы на странице «Контакт» таким образом, чтобы при потере фокуса заполняемого поля осуществлялась проверка корректности его заполнения.

В случае если поле заполнено корректно, оно должно быть подсвечено зеленым цветом, иначе оно должно быть подсвечено красным, а после данного поля должна появиться надпись, поясняющая характер ошибки. После исправления пользователем ошибки, надпись должна исчезнуть. Если все поля формы заполнены корректно, должна стать активной кнопка «Отправить».

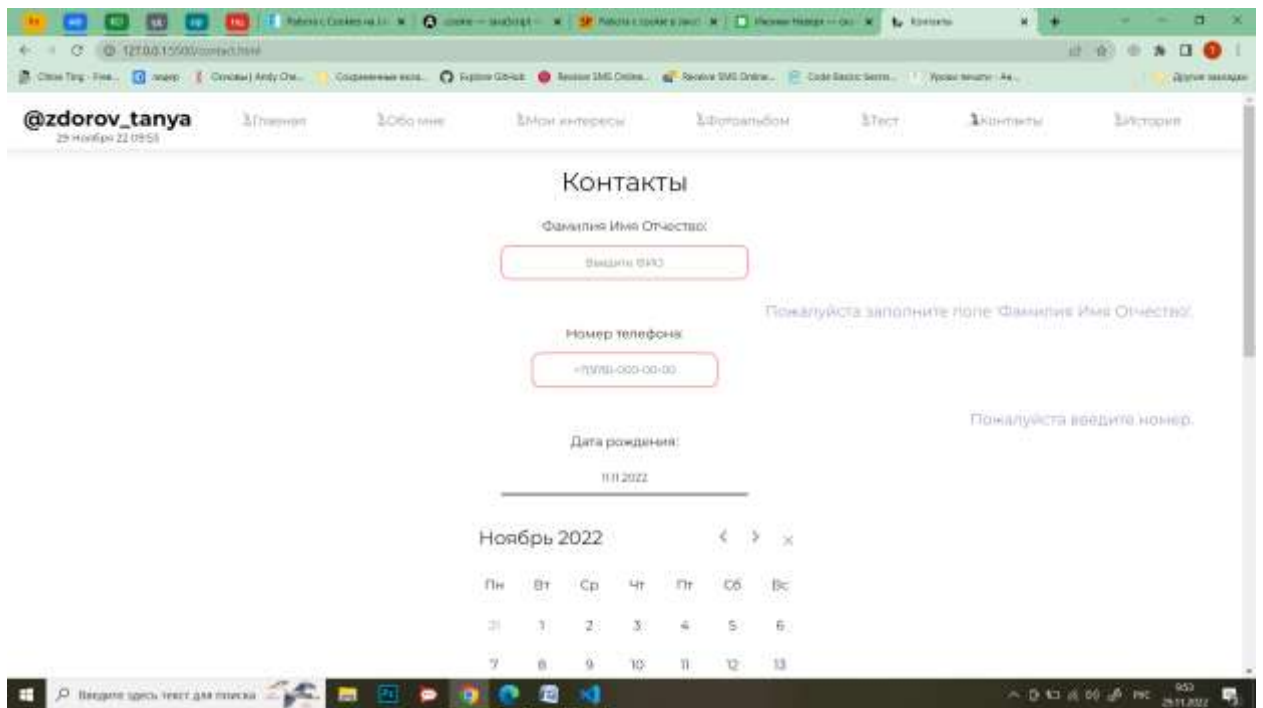


Рисунок 5 – Красная подсветка при некорректном заполнении полей

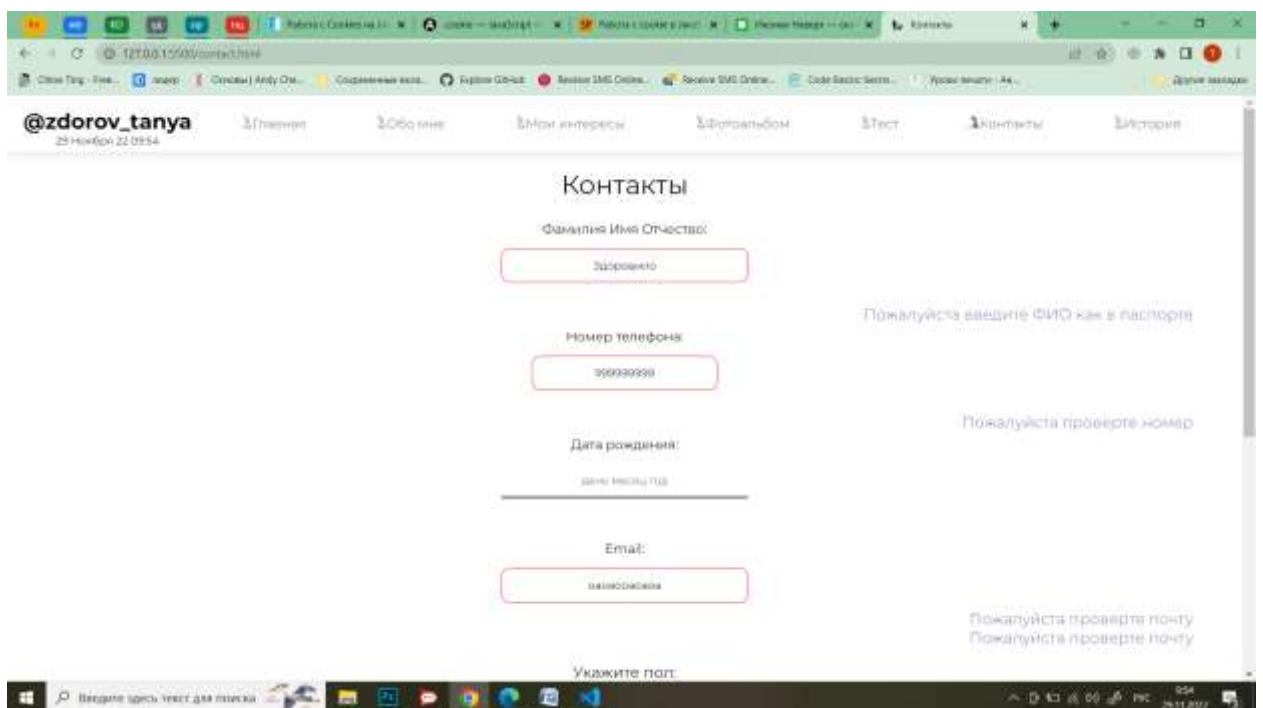


Рисунок 6 – Вывод сообщения об ошибке

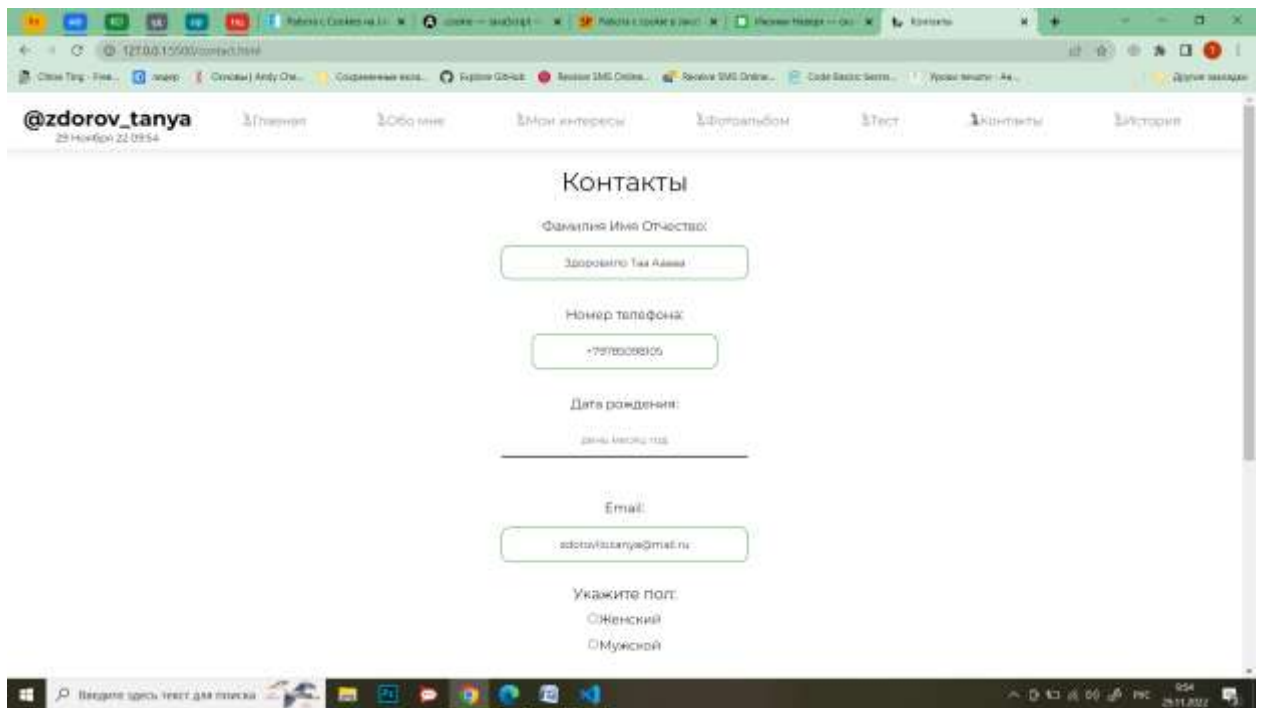


Рисунок 7 – Зеленая подсветка при корректном заполнении полей и удаление предыдущих сообщений об ошибках

Код программы:

```
function validate_form()
{
    if (!fio()) return false;

    if (!phone()) return false;

    if (!post()) return false;

    if (!check_radio()) return false;

    var word = document.contact_form.comment;
    if (!text_area(word)) return false;

    var brth = document.contact_form.daybirn;
    if (!bData(brth)) return false;

    return true;
}

function fio (){
    var uname = document.contact_form.contact_name.value;
    var valid = true;
    var id = document.querySelector('#Username');

    let mistake = document.getElementsByClassName("error");

    var i, d, text1;
```

```

d = document.createElement("div");
i = document.createElement("h3");
text1 = document.createTextNode("Пожалуйста заполните поле 'Фамилия Имя Отчество'.");
i.appendChild(text1);
d.appendChild(i);

var j, b, text2;

b = document.createElement("div");
j = document.createElement("h3");
text2 = document.createTextNode("Пожалуйста введите ФИО как в паспорте");
j.appendChild(text2);
b.appendChild(j);

if ((uname == "")){
    mistake[0].appendChild(d);
    b.remove();
    valid = false;
    id.style.border="1px solid red";
    id.style.borderRadius = "10px";
    // id.focus();

};

if (uname != ""){
    if (uname.match(/^[А-Я][а-я]+\s[А-Я][а-я]+\s[А-Я][а-я]+$/)){
        var del = mistake[0].childNodes.length;
        id.style.border="1px solid green";
        id.style.borderRadius = "10px";
        for(var i=del-1; i>=0; i--){
            mistake[0].childNodes[i].remove();
        }
        valid=true;
    }
    else {
        d.remove();
        mistake[0].appendChild(b);
        valid = false;
        id.style.border="1px solid red";
        id.style.borderRadius = "10px";
        // id.focus();
    }
};

button();

return valid;
}

let numberInput = document.getElementById("phone");
numberInput.addEventListener("blur", phone);

function phone (){
    var tel = document.contact_form.phone_num.value;
    var valid = true;
    var id = document.querySelector('#phone');

    let mistake = document.getElementsByClassName("error");

    var i, d, text1;

    d = document.createElement("div");
    i = document.createElement("h3");
    text1 = document.createTextNode("Пожалуйста введите номер.");
    i.appendChild(text1);

```

```

        d.appendChild(i);

        var j, b, text2;

        b = document.createElement("div");
        j = document.createElement("h3");
        text2 = document.createTextNode("Пожалуйста проверте номер");
        j.appendChild(text2);
        b.appendChild(j);

        if ((tel == "")){
            mistake[1].appendChild(d);
            b.remove();
            valid = false;
            id.style.border="1px solid red";
            id.style.borderRadius = "10px";
            // id.focus();

        };

        if (tel != ""){
            if (tel.match(/^\\+[73][0-9]{9,11}$/)){
                var del = mistake[1].childNodes.length;
                id.style.border="1px solid green";
                id.style.borderRadius = "10px";
                for(var i=del-1; i>=0; i--){
                    mistake[1].childNodes[i].remove();
                }
                valid=true;
            }
            else {
                d.remove();
                mistake[1].appendChild(b);
                valid = false;
                id.style.border="1px solid red";
                id.style.borderRadius = "10px";
                // id.focus();
            }
        };

        button();

        return valid;
    }

let postInput = document.getElementById("UserEmail");
postInput.addEventListener("blur", post);

function post (){
    var email = document.contact_form.email_name.value;
    var valid = true;
    var id = document.querySelector('#UserEmail');

    let mistake = document.getElementsByClassName("error");

    var i, d, text1;

    d = document.createElement("div");
    i = document.createElement("h3");
    text1 = document.createTextNode("Пожалуйста заполните поле 'Email'");
    i.appendChild(text1);
    d.appendChild(i);

    var j, b, text2;

    b = document.createElement("div");

```

```

j = document.createElement("h3");
text2 = document.createTextNode("Пожалуйста проверте почту");
j.appendChild(text2);
b.appendChild(j);

if ((email == "")){
    mistake[2].appendChild(d);
    b.remove();
    valid = false;
    id.style.border="1px solid red";
    id.style.borderRadius = "10px";
};

if (email != ""){
    if (email.match(/^[_a-z0-9-\+]+\.[_a-z0-9-]*@[a-z0-9-]+\.[a-z0-9-]*\.[a-z]{2,}$/i)){
        var del = mistake[2].childNodes.length;
        id.style.border="1px solid green";
        id.style.borderRadius = "10px";
        for(var i=del-1; i>=0; i--){
            mistake[2].childNodes[i].remove();
        }
        valid=true;
    }
    else {
        d.remove();
        mistake[2].appendChild(b);
        valid = false;
        id.style.border="1px solid red";
        id.style.borderRadius = "10px";
    }
};

button();

return valid;

}

// function fio (){
//     var uname = document.contact_form.contact_name.value;
//     var valid = true;
//     var id = document.querySelector('#Username');
//
//     if ( uname == "" )
//     {
//         alert ( "Пожалуйста заполните поле 'Фамилия Имя Отчество'." );
//         id.focus();
//         valid = false;
//     }
//
//     if (!uname.match(/^[A-Я][a-я]+\s[A-Я][a-я]+\s[A-Я][a-я]+$/)){
//         alert ( "Пожалуйста введите ФИО как в паспорте" );
//         id.focus();
//         valid = false;
//     }
//
//     return valid;
// }

// function phone (tel){
//     var valid = true;
//     var id = document.querySelector('#phone');
//
//     if ( tel == "" )

```

```

//      {
//          alert ( "Пожалуйста введите номер." );
//          id.focus();
//          valid = false;
//      }

//      if (!tel.match(/^+[73][0-9]{9,11}$/)) {
//          alert ( "Пожалуйста проверьте номер" );
//          id.focus();
//          valid = false;
//      }

//      return valid;
// }

// function post (email){
//     var valid = true;
//     var id = document.querySelector('#UserEmail');

//     if ( email == "" )
//     {
//         alert ( "Пожалуйста заполните поле 'Email'." );
//         id.focus();
//         valid = false;
//     }

//     return valid;
// }

function check_radio (){
    var valid = false;

    if(document.getElementById("w").checked || document.getElementById("m").checked){
        valid = true;
    }

    if(valid){
    }
    else{
        alert ("Укажите пол.");
    }

    return valid;
}

function text_area (word){
    var valid = true;

    if ( word.value == "" )
    {
        alert ( "Пожалуйста заполните текстовое поле." );
        word.focus();
        valid = false;
    }

    return valid;
}

function bData (brth){
    var valid = true;

    if ( brth.value == "" )
    {
        alert ( "Пожалуйста укажите дату рождения." );
        brth.focus();
        valid = false;
    }
}

```

```

    return valid;
}

function button (){
    var uname = document.contact_form.contact_name.value;
    var email = document.contact_form.email_name.value;
    var tel = document.contact_form.phone_num.value;

    if ((email.match(/^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.?[a-z]{2,})$/i)) &&
        (tel.match(/^\+[73][0-9]{9,11}$/)) && (uname.match(/^[A-Я][a-я]+\s[A-Я][a-я]+\s[A-Я][a-я]+$/)))
    {document.getElementById("btn").disabled = false} else {document.getElementById("btn").disabled = true};
}

```

5. Реализовать открытие в динамически формируемом новом окне (блоке DIV) соответствующих больших фото при щелчке мыши по маленьким фото на странице «Фотоальбом».

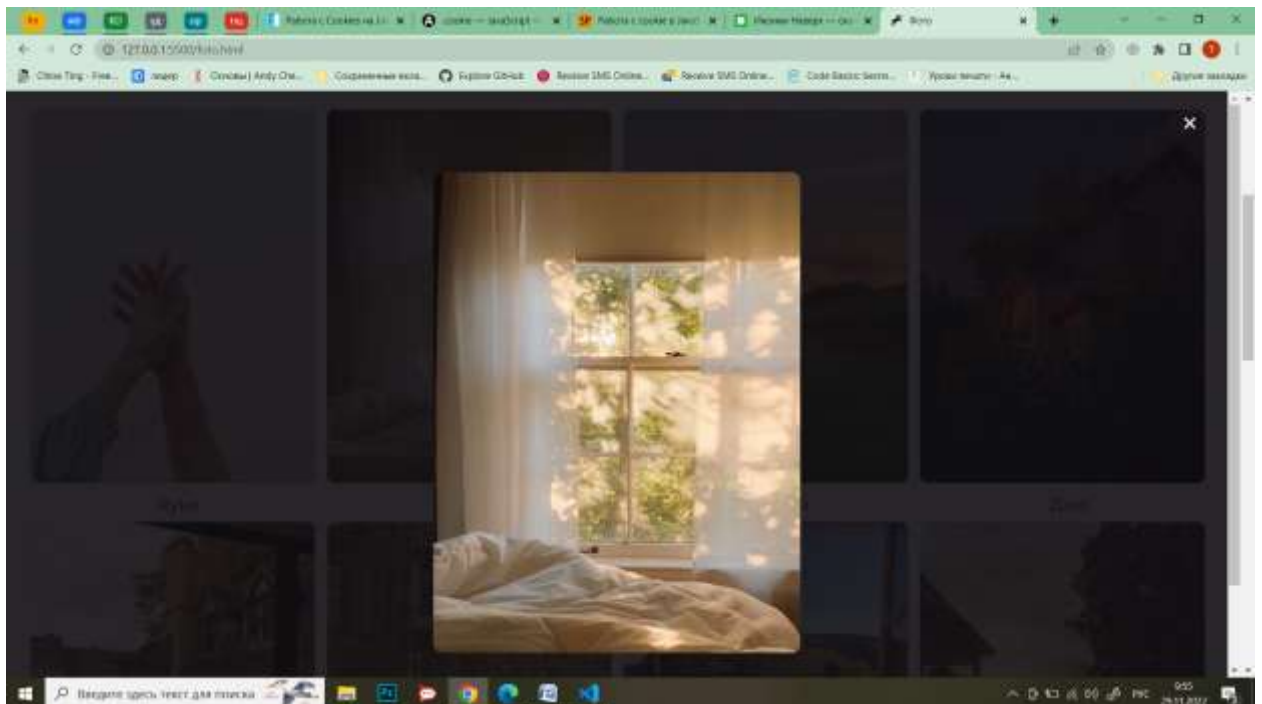


Рисунок 8 – Открытие большой фотографии по щелчку

Код программы:

```

window.onload = function() { //скрипт выполнен после загрузки страницы
    var fotos = new Array(15);
    fotos[0]="./img/heand.jpg";
    fotos[1]="./img/room.jpg";
    fotos[2]="./img/girl.jpg";
    fotos[3]="./img/house.jpg";
    fotos[4]="./img/book.jpg";
    fotos[5]="./img/rain.jpg";
    fotos[6]="./img/window.jpg";
    fotos[7]="./img/dog.jpg";
    fotos[8]="./img/sheep.jpg";

```

```

    fotos[9]="./img/dour.jpg";
    fotos[10]="./img/cow.jpg";
    fotos[11]="./img/grass.jpg";
    fotos[12]="./img/milka.jpg";
    fotos[13]="./img/wilage.jpg";
    fotos[14]="./img/pirs.jpg";

var titles = new Array(15);
    titles[0]="какие-то руки";
    titles[1]="комната, выходящая на солнечную сторону";
    titles[2]="бежит куда-то";
    titles[3]="ну прям голден ауер";
    titles[4]="хотела бы я прочитать столько книг";
    titles[5]="а ведь кто-то только что помыл машину";
    titles[6]="у меня вид на детскую площадку";
    titles[7]="пёсель";
    titles[8]="интересно, о чем думает овечка";
    titles[9]="кажется тут нужен капитальный ремонт";
    titles[10]="вы коровок видите? а они есть";
    titles[11]="просто травка";
    titles[12]="в день корова может давать до 20 литров молока";
    titles[13]="неплохое место для отдыха";
    titles[14]="я не умею плавать";

var text = new Array(15);
    text[0]="Руки";
    text[1]="Комната";
    text[2]="Девушка";
    text[3]="Дом";
    text[4]="Книги";
    text[5]="Дождь";
    text[6]="Окно";
    text[7]="Собака";
    text[8]="Овечки";
    text[9]="Дверь";
    text[10]="Коровки";
    text[11]="Луг";
    text[12]="Милка";
    text[13]="Деревня";
    text[14]="Пирс";

var all = document.createElement("div");
for (var i = 0; i < 15; i++)
{
    var d = document.createElement("div");
    var img = document.createElement("img");
    img.id = 'myImg';
    var word = document.createElement("h3");

    var elemText = document.createTextNode(text[i]);

    d.classList.add('foto');
    img.src = fotos[i];
    img.title = titles[i];

    // console.log(text[i] +"p");
    img.onclick = function(){
        modal.style.display = "block";
        modalContent.src = this.src;
        write = document.createTextNode("поооппоо");
    }

    word.appendChild(elemText);
    d.appendChild(img); //добавляет новые html элементы к существующим
    d.appendChild(word);
    all.appendChild(d);
}

```



```

all.classList.add('col-md-4');
document.body.appendChild(all);

var m = document.createElement("div");
for (var i = 0; i < 15; i++)
{

    var modal = document.createElement("div");
    modal.id = 'myModal';
    modal.classList.add('modal');

    var close = document.createElement("span");
    close.classList.add('close');
    close.innerHTML = "&times;";

    var modalContent = document.createElement("img");
    modalContent.id = "img01";
    modalContent.classList.add('modal-content');

    var write = document.createElement("div");
    write.id = 'caption';
    write.classList.add('caption');

    close.onclick = function() {
        modal.style.display = "none";
    }

    modal.appendChild(modalContent);
    modal.appendChild(write);
    modal.appendChild(close);
    m.appendChild(modal);

}
document.body.appendChild(m);
};

```

6. Добавить страницу «История просмотра». На данной странице реализовать отображение двух таблиц:

«История текущего сеанса»

— в данной таблице отображается количество посещений каждой страницы за время текущего сеанса. Реализовать хранение этих данных в Local Storage.

История текущего сеанса

Главная 2

Фотоальбом 8

История 2

Обо мне 2

Контакты 2

Мои интересы 1

Мои интересы 1

Тест 0

Рисунок 8 – История текущего сеанса

Код программы:

```
function cookieWithoutMilk(){
    sessionStorage.setItem(window.location.href, Number(sessionStorage.getItem(window.location.href)) + Number(1));
    console.log(sessionStorage.getItem(window.location.href));
};
cookieWithoutMilk();

function forHistory(){
    const pages = ["index", "call", "interest", "foto", "test", "contact", "history"];

    let name = document.createElement("div");
    name.innerText = "История текущего сеанса";
    name.classList.add("todayhistory");

    document.body.appendChild(name);

    for (let i=0; i<sessionStorage.length; i++){
        for (let j =0; j<pages.length; j++){
            if(sessionStorage.key(i).includes(pages[j])){
                switch (pages[j]){
                    case "index":
                        let indexText = document.createElement("div");
                        indexText.id = "indexId";
                        indexText.innerText = "Главная " +
sessionStorage.getItem("http://127.0.0.1:5500/index.html");

                        name.appendChild(indexText);

                        break;

                    case "call":
                        let callText = document.createElement("div");
                        callText.id = "callId";
                        callText.innerText = "Обо мне " +
sessionStorage.getItem("http://127.0.0.1:5500/call.html");

                        name.appendChild(callText);
```

```

        break;

        case "interest":

            let interestText = document.createElement("div");
            interestText.id = "interestId";
            interestText.innerHTML = "Мои интересы " +
sessionStorage.getItem("http://127.0.0.1:5500/interest.html");

            name.appendChild(interestText);

            break;

        case "foto":

            let fotoText = document.createElement("div");
            fotoText.id = "fotoId";
            fotoText.innerHTML = "Фотоальбом " +
sessionStorage.getItem("http://127.0.0.1:5500/foto.html");

            name.appendChild(fotoText);

            break;

        case "test":

            let testText = document.createElement("div");
            testText.id = "testId";
            testText.innerHTML = "Тест " +
sessionStorage.getItem("http://127.0.0.1:5500/test.html");

            name.appendChild(testText);

            break;

        case "contact":

            let contactText = document.createElement("div");
            contactText.id = "contactId";
            contactText.innerHTML = "Контакты " +
sessionStorage.getItem("http://127.0.0.1:5500/contact.html");

            name.appendChild(contactText);

            break;

        case "history":

            let historyText = document.createElement("div");
            historyText.id = "historyId";
            historyText.innerHTML = "История " +
sessionStorage.getItem("http://127.0.0.1:5500/history.html");

            name.appendChild(historyText);

            break;
    };
}

}

}

if (document.getElementById("indexId") == null ){
    let indexText = document.createElement("div");
    indexText.id = "indexId";
    indexText.innerHTML = "Главная " + "0";

    name.appendChild(indexText);

```

```

}

if (document.getElementById("callId") == null ){
    let callText = document.createElement("div");
    callText.id = "callId";
    callText.innerText = "Обо мне " + "0";

    name.appendChild(callText);
}

if (document.getElementById("interestId") == null ){
    let interestText = document.createElement("div");
    interestText.id = "interestId";
    interestText.innerText = "Мои интересы " + "0";

    name.appendChild(interestText);
}

if (document.getElementById("fotoId") == null ){
    let fotoText = document.createElement("div");
    fotoText.id = "fotoId";
    fotoText.innerText = "Фотоальбом " + "0";

    name.appendChild(fotoText);
}

if (document.getElementById("testId") == null ){
    let testText = document.createElement("div");
    testText.id = "testId";
    testText.innerText = "Тест " + "0";

    name.appendChild(testText);
}

if (document.getElementById("contactId") == null ){
    let contactText = document.createElement("div");
    contactText.id = "contactId";
    contactText.innerText = "Контакты " + "0";

    name.appendChild(contactText);
}

if (document.getElementById("historyId") == null ){
    let historyText = document.createElement("div");
    historyText.id = "historyId";
    historyText.innerText = "История " + "0";

    name.appendChild(historyText);
}
}

```

«История за все время»

— в данной таблице отображается количество посещений каждой страницы за все время. Реализовать хранение этих данных в хранилище Cookies. Для выполнения этого задания необходимо создать два JavaScript метода: `getCookie (name)` и `setCookie(name, value, expiration_days)`.

История за все время

Главная

9

Обо мне

5

Мои интересы

3

Фотоальбом

1

Тест

9

Контакты

1

История

1

Рисунок 9 – История за все время

Код программы:

```
const pages=["Главная", "Обо мне", "Мои интересы", "Фотоальбом", "Тест", "Контакты", "История"]

function createCookie(){

    var locate = (!window.location.href.includes("interest")) ? window.location.href :
"http://127.0.0.1:5500/interest.html";
    console.log("locate - " + locate);

    if (!document.cookie.includes(locate)){
        document.cookie = locate + "=1";
        console.log("new");
        return;
    }

    var ind = document.cookie.indexOf(locate);//проверяет вхождение в строку
    var si = Number(document.cookie.indexOf(";", ind)) + Number(1);
    var cookies = document.cookie.slice(ind, si);

    ind = Number(cookies.indexOf("=")) + Number(1);
    si = Number(cookies.indexOf(";", ind));
    cookies = cookies.slice(ind, si);

    cookies = Number(cookies) + Number(1);

    document.cookie = locate + "=" + cookies; //через неделю после посещения страницы данные удаляться, добавить
время жизни?
    console.log("old");
}

function getCookie(){

    let menu = document.querySelectorAll(".forPicture");
    let cookDiv = document.createElement("div");
    let histDiv = document.createElement("div");
    let metrDiv = document.createElement("div");

    histDiv.innerText = "История за все время";
    cookDiv.appendChild(histDiv);
```

```

cookDiv.classList.add("todayhistory");
console.log("11111");

let countvisit;
let doccookie = document.cookie + ";";

for (let i=0; i<7; i++){

    // let locate = (i != 2) ? menu[i].href : "http://127.0.0.1:5500/interest.html";
    let locate = menu[i].href;

    if (!doccookie.includes(locate)){
        countvisit = 0;
    }
    else{
        let ind = doccookie.indexOf(locate);
        let si = Number(doccookie.indexOf(";", ind)) + Number(1);
        let cookies = doccookie.slice(ind, si);

        ind = Number(cookies.indexOf("=")) + Number(1);
        si = Number(cookies.indexOf(";", ind));
        countvisit = cookies.slice(ind, si);

    }

    var pageDivC = document.createElement("div");
    var countDivC = document.createElement("div");

    pageDivC.innerText = pages[i];
    countDivC.innerText = countvisit;

    metrDiv.appendChild(pageDivC);
    metrDiv.appendChild(countDivC);

}
cookDiv.appendChild(metrDiv);
document.body.appendChild(cookDiv);
}

createCookie();

```

Вывод:

В ходе выполнения лабораторной работы была исследована структура модели документа DOM, изучена динамическая объектная модель документа, предоставляемая стандартом DOM и системой событий языка JavaScript, возможность хранения данных на стороне клиента.

А так же были приобретены практические навыки работы с событиями JavaScript, деревом документа, Local Storage и Cookies.