# ЛАБОРАТОРНАЯ РАБОТА №11 «Исследование возможностей асинхронного взаимодействия с сервером. Технология AJAX»

## Цель работы

Исследовать возможности асинхронного обмена данными с сервером, изучить принципы асинхронной работы с сервером, приобрести практические навыки использования технологии AJAX для организации отправки данных на сервер и получения данных без полной перезагрузки страницы

## Порядок выполнения работы

1. На странице «Регистрация пользователя» реализовать проверку занятости логина без перезагрузки страницы (использовать событие onBlur или кнопку «Проверить занятость»). Использовать для отправки данных на сервер метод и формат данных, указанный в колонке «Задание №1» таблицы 4.3 в соответствии с вариантом задания.

2. Реализовать возможность комментирования записей блога для авторизированных пользователей сайта. В случае, если пользователь авторизирован отображать после каждой записи блога кнопку или ссылку «Добавить комментарий». При нажатии кнопки/ссылки отображать модальное окно (реализовать с использванием, например, с использованием абсолютно спозиционированного блока div), содержащее поле ввода текста комментария и кнопку «Отправить». Реализовать отправку комментария на сервер и его отображение (дата и время, автор, текст комментария) под записью блога (или под уже существующими комментариями данной записи) без полной перезагрузки страницы. Комментарии хранить в специально разработанной таблице базы данных. Использовать для отправки данных на сервер метод и формат данных, указанный в колонке «Задание №2» таблицы 4.3 в соответствии с вариантом задания.

3. В зоне администратора реализовать возможность редактирования записей блога. Отображать возле каждой записи блога кнопку или ссылку «Изменить». При нажатии кнопки/ссылки отображать окно (div), содержащее поле ввода с темой записи блога и поле ввода с текстом записи блога для редактирования, а также кнопку/ссылку «Сохранить изменения». Реализовать отправку данных и их отображение без полной перезагрузки страницы. Использовать для отправки данных на сервер метод и формат данных, указанный в колонке «Задание №3» таблицы 4.3 в соответствии с вариантом задания.

## Ход работы

### На рисунке 4.1 представлена ошибка, которая возникает, когда пользователь вводит логин, который уже существует



Рисунок 4.1 – Ошибка при вводе существующего логина

### Если пользователь не авторизован он увидит сообщение, представленное на рисунке 4.2. При входе пользователь увидит кнопку «добавить комментарий» (Рисунок 4.3). Нажав на неё появится модально окно (Рисунок 4.4). После отправки данных пользователь сможет увидеть добавленный комментарий (Рисунок 4.5)

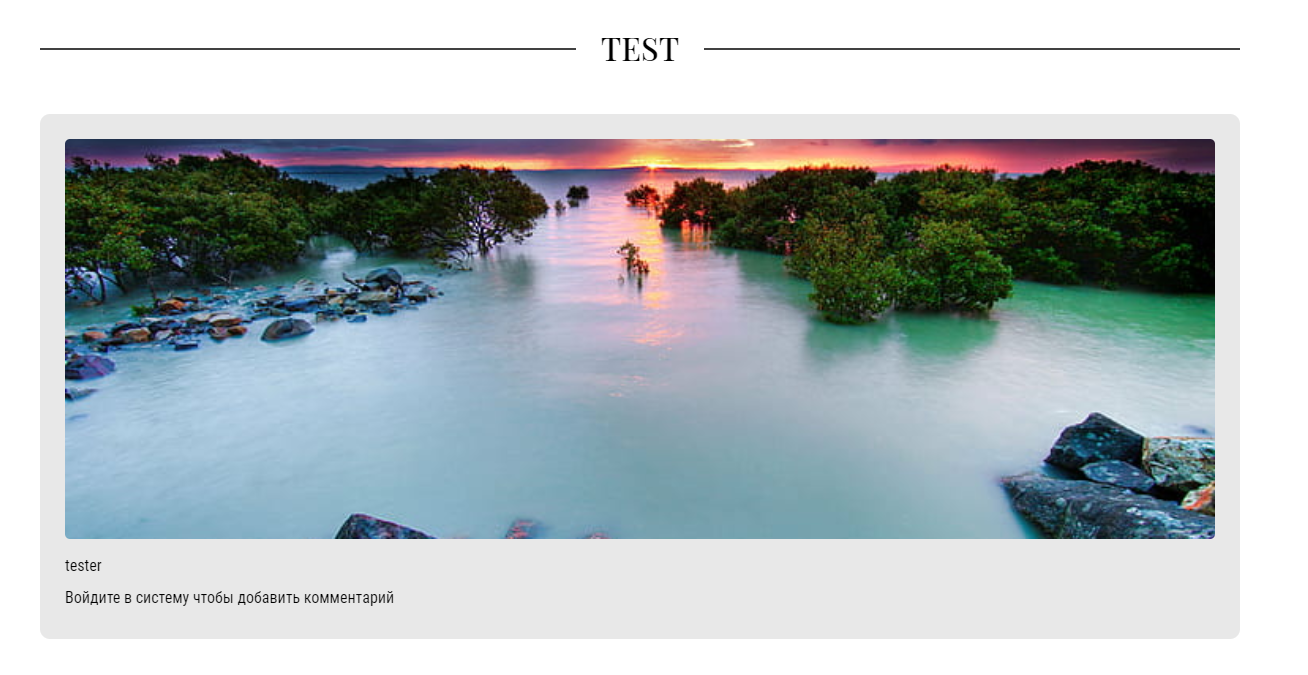


Рисунок 4.2 – Сообщение о необходимости входа

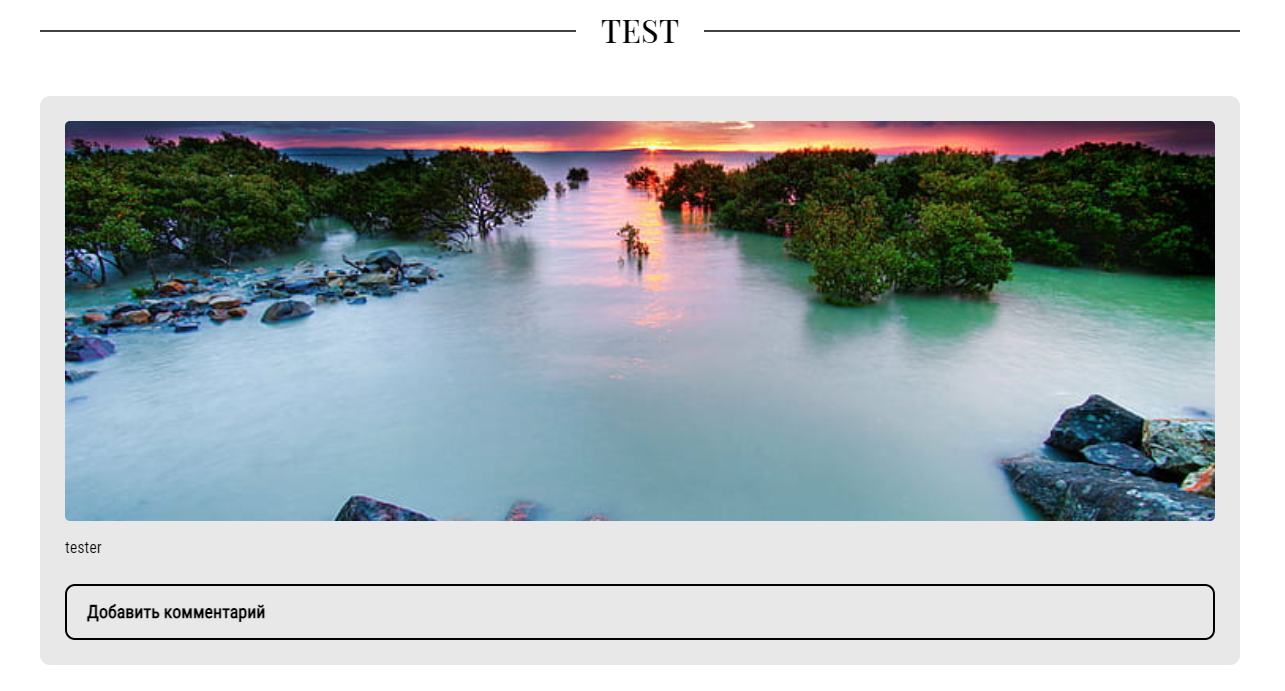


Рисунок 4.3 – Кнопка «Добавить комментарий»

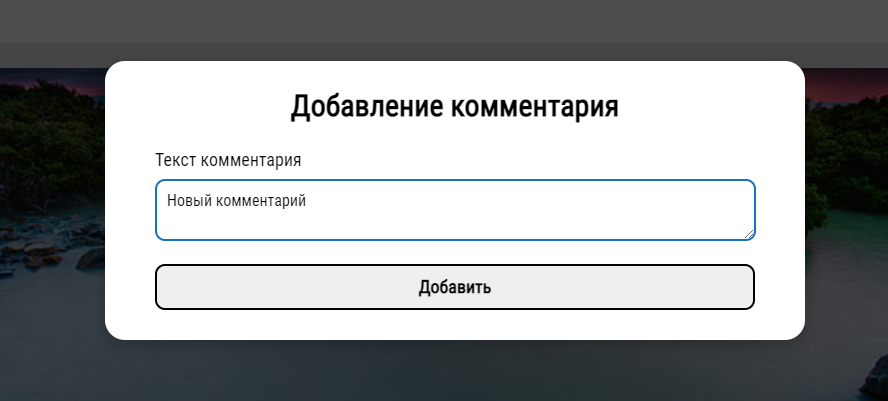


Рисунок 4.4 – Модально окно

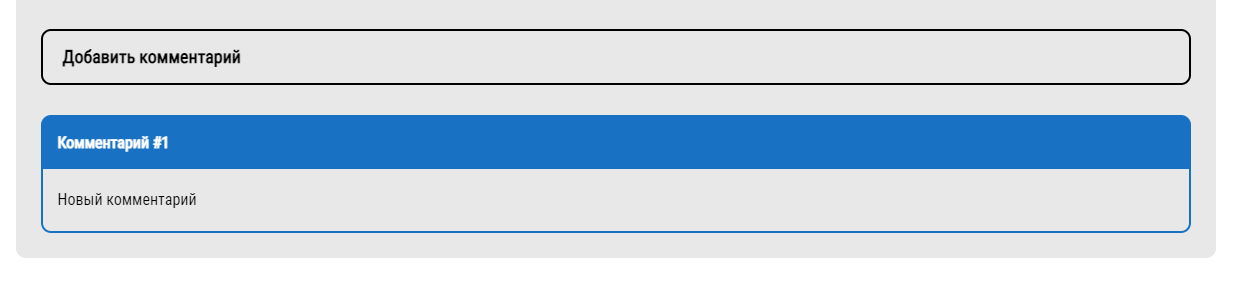


Рисунок 4.5 – Добавленный комментарий

### На рисунке 4.6 показана кнопка для редактирования записи блога. При клике на неё появится модально окно (Рисунок 4.7). Кликнув на кнопку изменить тема записи и текст сообщения появится сообщение об успешном изменении (Рисунок 4.8), а также отобразятся введённые данные(Рисунок 4.9).



Рисунок 4.6 – Кнопка для редактирования записи блога

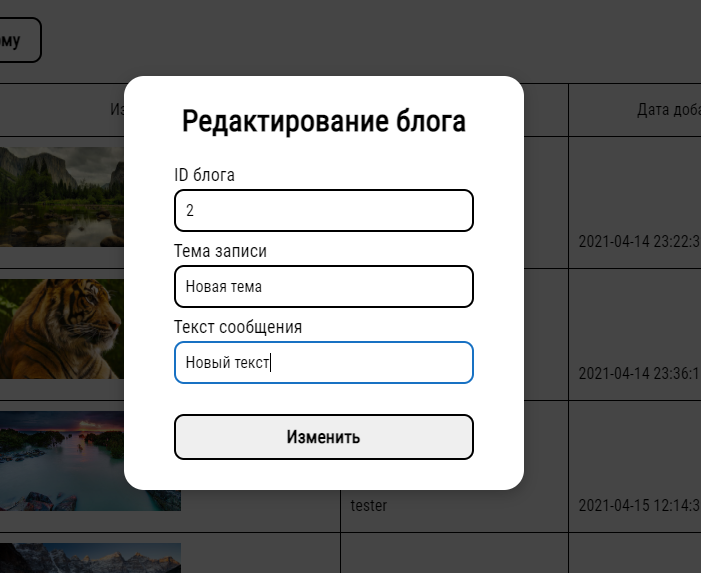


Рисунок 4.7 – Модальное окно для редактирования записи блога

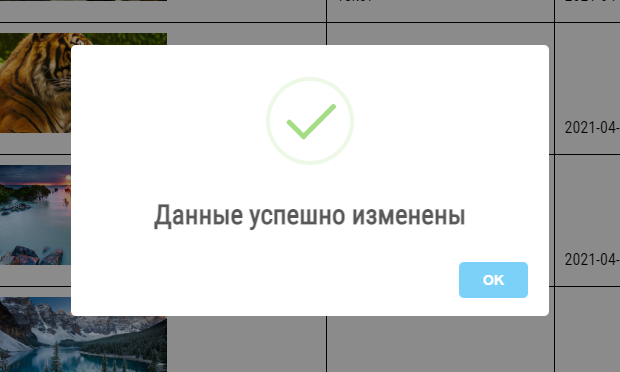


Рисунок 4.8 – Сообщение об успешном редактировании



Рисунок 4.9 – Отредактированные данные

# Вывод

В ходе выполнения данной лабораторной работы были исследованы возможности асинхронного обмена данными с сервером, изучены принципы асинхронной работы с сервером, приобретены практические навыки использования технологии AJAX для организации отправки данных на сервер и получения данных без полной перезагрузки страницы. Кроме этого реализован функционал проверки занятости логина, добавления комментария и редактирования записи без перезагрузки страницы с помощью технологии ajax.

# Приложение А

Файл public/js/index.js

// ============Подменю===========  
function showSubMenu(e) {  
 ***console***.log("Ok")  
 if ($(this).children().length > 1) {  
 $(this).find(".sub-menu").css("display", "grid");  
 } else {  
 return false;  
 }  
}  
  
function hideSubMenu(e) {  
 if ($(this).children().length > 1) {  
 $(this).find(".sub-menu").css("display", "none");  
 } else {  
 return false;  
 }  
}  
  
  
// ============Время============  
let ***month*** = "";  
  
let ***russianMonthName*** = ["январь", "февраль", "март", "апрель", "май", "июнь", "июль", "август", "сентябрь", "октябрь", "ноябрь", "декабрь"]  
  
function drawTime() {  
 let date = new ***Date***();  
  
 let h = date.getHours();  
 let m = date.getMinutes();  
 let s = date.getSeconds();  
  
 s < 10 ? s = "0" + s : s;  
 m < 10 ? m = "0" + m : m;  
 h < 10 ? h = "0" + h : h;  
  
 time.innerHTML = `${h}:${m}:${s} ${date.getDate()} ${***russianMonthName***[date.getMonth()]} ${date.getFullYear()}`;  
}  
  
let ***timerId*** = setInterval(() => {  
 drawTime();  
}, 1000);  
  
$(***document***).ready(function () {  
 let thisHistory, globalHistory;  
  
 let pathname = ***document***.location.pathname; // адрес текущей страницы  
 pathname = pathname.slice(pathname.lastIndexOf("/"), pathname.length);  
  
 drawTime();  
 saveThisHistory(pathname);  
 saveGlobalHistory(pathname);  
 if (pathname === "/history") drawHistoryTable();  
  
 $("#menu li").each(function () {  
 let thisURL = $(this).children().first();  
  
 if (thisURL.attr("href") === pathname) {  
 thisURL.addClass("active");  
 }  
 })  
})  
  
// =================История просмотра================  
function saveThisHistory(pathname) {  
 if (***sessionStorage***.getItem('thisHistory') === null) {  
 thisHistory = {};  
 ***sessionStorage***.setItem('thisHistory', ***JSON***.stringify(thisHistory));  
 }  
  
 thisHistory = ***JSON***.parse(***sessionStorage***.getItem('thisHistory'));  
 thisHistory[pathname] === undefined ? thisHistory[pathname] = 1 : thisHistory[pathname]++;  
 ***sessionStorage***.setItem('thisHistory', ***JSON***.stringify(thisHistory));  
}  
  
function saveGlobalHistory(pathname) {  
 if (***localStorage***.getItem('globalHistory') === null) {  
 globalHistory = {};  
 ***localStorage***.setItem('globalHistory', ***JSON***.stringify(globalHistory));  
 }  
  
 globalHistory = ***JSON***.parse(***localStorage***.getItem('globalHistory'));  
 globalHistory[pathname] === undefined ? globalHistory[pathname] = 1 : globalHistory[pathname]++;  
 ***localStorage***.setItem('globalHistory', ***JSON***.stringify(globalHistory));  
}  
  
// Рисуем таблицу для истории  
function drawHistoryTable() {  
 $("#menu>li").each(function () {  
 let link = $(this).children().first();  
 let linkSrc = link.attr("href");  
  
 $("#history\_table").children()[0].insertAdjacentHTML("afterend",  
 `<tr>  
 <td>${link.text()}</td>  
 <td>${thisHistory[linkSrc] || "0"}</td>  
 <td>${globalHistory[linkSrc] || "0"}</td>  
 </tr>  
 `);  
 });  
}  
  
function drawActiveImg(idActiveImg, firstOpen) {  
 if (idActiveImg <= 0) {  
 $(".prev-img").addClass("disabled");  
 } else $(".prev-img").removeClass("disabled");  
  
 if (idActiveImg >= 14) {  
 $(".next-img").addClass("disabled");  
 } else $(".next-img").removeClass("disabled");  
  
 let srcClickedImg = $(`.foto\_img[data-img-id=${idActiveImg}]`).attr("src");  
 let textClickedImg = $(`.foto\_img[data-img-id=${idActiveImg}]`).attr("title");  
  
 $(".num-this-img").text(idActiveImg + 1);  
  
 if (firstOpen) {  
 $("#album\_img img").prop("src", srcClickedImg);  
 $("#album\_img>span").text(textClickedImg);  
 } else {  
 $("#album\_img>span").fadeOut();  
 $("#album\_img img").fadeOut(400, "swing", function () {  
 $("#album\_img img").prop("src", srcClickedImg).fadeIn();  
 $("#album\_img>span").text(textClickedImg).fadeIn();  
 }, false);  
 }  
  
}  
  
function attachScript(id, src) {  
 let element = ***document***.createElement("script")  
 element.src = src  
 element.id = id  
 ***document***.getElementsByTagName("head")[0].appendChild(element)  
}  
  
function sendXHRRequset(method, url, body = null) {  
 return new ***Promise***((resolve, reject) => {  
 let xhr = new ***XMLHttpRequest***()  
  
 xhr.open(method, url)  
 xhr.responseType = "text"  
  
 xhr.onload = () => {  
 if (xhr.status >= 400) {  
 reject(xhr.response)  
 } else resolve(xhr.response)  
 }  
  
 xhr.onerror = () => {  
 reject(xhr.response)  
 }  
  
 xhr.send(body)  
 })  
}  
  
function sendFetchRequest(method, url, body = null) {  
 const headers = {}  
  
 return fetch(url, {  
 method: method,  
 body: body,  
 headers: headers  
 }).then((response) => {  
 if (response.ok) {  
 return response.json()  
 }  
  
 return response.json().catch((error) => {  
 const e = new ***Error***("Что-то пошло не так")  
 e.data = error  
 throw e  
 })  
 })  
}  
  
function formattingFormData(\_this) {  
 let formData = new ***FormData***(\_this[0]);  
 let error = [];  
 let supportedFormatsImg = ["image/png", "image/jpg", "image/jpeg"];  
  
 formData.forEach((item, i) => {  
 if (typeof item === "object") {  
 if (supportedFormatsImg.includes(item.type)) {  
 formData.set(i.toString(), item.name);  
 } else {  
 error.push("Неверный формат файла");  
 }  
 }  
 })  
  
 if (error.length > 0) {  
 swal({  
 title: "Ошибка",  
 text: error[0],  
 icon: "error",  
 });  
 return false;  
 } else return formData;  
}  
  
function showModal(modal = $(".modal")) {  
 modal.addClass("active")  
 $(".black-bg").addClass("active")  
}  
  
function hideModal() {  
 $(".modal").removeClass("active")  
 $(".black-bg").removeClass("active")  
}  
  
function checkLogin(loginIsBusy){  
 if( loginIsBusy )  
 $(".notification\_\_item.login-busy").css("display", "block")  
  
 setTimeout(() => {  
 $(".notification\_\_item.login-busy").css("display", "none")  
 }, 7000)  
}  
  
$(***document***).ready(function () {  
 $(".menu-item").on("mouseenter", showSubMenu);  
 $(".menu-item").on("mouseleave", hideSubMenu);  
  
 $(".next-img").on("click", function (e) {  
 let activeImg = $("#album\_img").data("active-img");  
 $("#album\_img").data("active-img", activeImg + 1);  
 drawActiveImg(activeImg + 1, false);  
 })  
  
 $(".prev-img").on("click", function (e) {  
 let activeImg = $("#album\_img").data("active-img");  
 drawActiveImg(activeImg - 1, false);  
 $("#album\_img").data("active-img", activeImg - 1);  
 })  
  
  
 $(***document***).on("click", ".foto\_img", function (e) {  
 let activeImg = $(this).data("img-id");  
 $("#album\_img").data("active-img", activeImg);  
  
 drawActiveImg(activeImg, true);  
 $(".wrapper\_album\_img").toggleClass("active");  
 });  
  
 $("#black\_bg, .close").on("click", function () {  
 $(".wrapper\_album\_img").toggleClass("active");  
 });  
  
 $(".admin-controller\_\_btn").on("click", function () {  
 $(".admin-controller\_\_panel").toggleClass("active");  
 })  
  
 $(".edit-blog").on("click", function () {  
 let blogID = $(this).parents("tr").data("id")  
  
 showModal()  
 $(".blog-id").val(blogID)  
 })  
  
 $(".blog-comment").on("click", function () {  
 showModal()  
 })  
  
 $(".edit-blog-modal form").on("submit", function (e) {  
 e.preventDefault()  
 let blogID = $(this).find(".blog-id").val()  
 let url = $(this).attr("action")  
 let method = $(this).attr("method")  
 let body = formattingFormData($(this))  
  
 sendFetchRequest(method, url, body)  
 .then((data) => {  
 swal({  
 title: data.title,  
 icon: data.icon,  
 })  
 let editedBlog = $(".blog-edit-table").find(`tr[data-id='${blogID}']`)  
 editedBlog.find(".blog-title").text(data.blogTitle)  
 editedBlog.find(".blog-text").text(data.blogText)  
 hideModal()  
 })  
 .catch((err) => {  
 swal({  
 title: err.title,  
 icon: err.icon,  
 })  
 })  
 })  
  
 $("input[name='last\_comment\_num']").val($(".comments\_\_num span").eq(-1).text())  
  
 $(".blog-comment-modal form").on("submit", function (e) {  
 e.preventDefault()  
 let url = $(this).attr("action")  
 let method = $(this).attr("method")  
 let body = formattingFormData($(this))  
  
 sendXHRRequset(method, url, body)  
 .then((data) => {  
 hideModal()  
 $(".comments").append(data)  
 })  
 .catch((err) => {  
 swal({  
 title: err.title,  
 icon: err.icon,  
 })  
 })  
 })  
  
 function attachScript(id, src){  
 var element = ***document***.createElement("script")  
 element.src = src  
 element.id = id  
 ***document***.getElementsByTagName("head")[0].appendChild(element)  
 }  
  
 $(".register-form input[name='login']").on("blur", function (e) {  
 attachScript("search-login", "/user/checkLogin?login=" + $(this).val() )  
 });  
  
 $(".black-bg").on("click", hideModal)  
})

Файл app/models/CommentsModel.php

<?php  
  
namespace app\models;  
  
use app\core\BaseActiveRecord;  
use app\core\Model;  
  
class CommentsModel extends BaseActiveRecord  
{  
 protected static *$tablename* = 'comments';  
  
 public $id;  
 public $comment\_text;  
 public $blog\_id;  
  
 public function \_\_construct()  
 {  
 parent::*\_\_construct*();  
 }  
}