Лабораторна робота №6

Система управління датами бронювання та резервування

**1. Мета та завдання**

Вивчення основ розробки складних систем для обліку резервування та бронювання. Розробка алгоритмів взаємодії з користувачем, розробка сладних алгоритмів

**2. Короткі теоретичні відомості**

Приклад виконання простого Ajax запиту

$.ajax({

url: "url PHP script", // url to PHP file

beforeSend: function() { // function do something before sand request

// add preloader or else

}

})

.done(function( data ) { // function do something after data received

// data – is result of request

});

Виконання запиту відбувається в два етапи, спочатку виконується функція що опрацьовується до відправки запиту і функція при отриманні результату. Функція до виконання наприклад може бути призначення для відображення анімації загрузки контенту чи скрити попередній.

**Формування на розбірка інформації у форматі JSON.**

Розпаковка результату виконання PHP скрипта

$.ajax({

type: "POST",

url: " url PHP script",

data: {some data} // example id:1

success: function(data){

var res = JSON.parse(data); // JSON result

}

});

Запаковування даних в JSON строку

$data=array(

"id"=>1,

"name"=>$name,

);

echo json\_encode($data);

**Селектор елемента.**

Існує багато методів селектору елементів на сайті. Всі вони базуються на CSS вибірці. Приклад:

Вибірка елементу по id $(‘#id name');

Вибірка елементів по class $(‘.Class name’)

Складна вибірка $(‘.Class name>span[name=”name”]’); Вибірка всіх span елементів що наслідуються класом з параметром name і значенням name

**Обробка даних форми**

$(‘form’).serialize(); - поверне значення всіх полей у форматі для відправки запиту на PHP скрипт

Приклад JS: var data=$(form).serialize(); Результат: name=’name’&id=’id’ і т.д

При необхідності можливо додати свої дані, аналогічно роботі з простою строкою

Приклад: data=data+’&newdata=”information”’

**Обробка даних на стороні сервера**

foreach($\_POST){}

foreach($\_GET){}

Формат даних що приселається на сервер залежить від вказаного типу даних в запиті Ajax через параметр type: ‘POST’ або ‘GET’

**Заміна даних на сайті без перезавантаження сторінки**

Існує декілька методів роботи з даними на сайті, серед яких, заміна інформації на нову, добавлення інших даних, добавлення з переду, з заду та ін.

Приклад.

$(‘.content’).append(data); - добавлення інформації після всьго контенту в елементі з імям класу content

$(‘.content’).html(‘’); - знищити, або замінити на пусту строку всю інфорацію всередині контейнера.

var html=$(‘.content’).html(); - отримання всієї інформації з контейнера

html+data;$(‘.content’).html(html); - замінити дані тими які були з добавленням нових

Як видно існує багато методів роботи з наповненням сайту

Якщо виконувати роботу з таблицями необхідно памятати про всю структуру таблиці. Добавлення нових строк буде виглядати так:

new\_tr=’<tr><td></td></tr>’;

$(‘.table>tbody’).append(new\_tr);

**3. Завдання**

Виконати просту верстку дизайну сайту з налаштування адаптивних версій на 500px та 700px. Розробити динамічну зміну інформації без перезагрузки сторінки за допомогою Ajax запитів та зміни контенту на JS. Організувати взаємодію зміни, добавлення та редагування даних в БД.

Загальний вигляд сайту

Верх сайту. Містить логотип та фон

Тіло сайту – інформація з таблицю даних резерву БД

Низ сайту. Містить форму добавлення даних

Форма пошуку

Форма пошуку містить такі поля як «Дата з» «Дата до» «Статус» та кнопку відправлення

При заповненні форми та відпраці необхідно написати скрипт на PHP який буде перевіряти отримані дані і повертати результат по кожному полю.

Якщо результат не містить помилок то відправити запит на отримання даних таблиці резервів з БД по отриманих фільтрам, параметрам

Структура блоку добавлення інформації

Дата з

Дата до

Статус checker

Додати/Пошук

Алгоритм роботи сайту

Search.php

Add.php

index.php

Script.js

style.css

Parse.php

Container

Array of search result from

Add form

Search form

status.php

**Пошук.** Введення даних, перевірка відправлених даних у випадку відсутності помилки відправити запит на отримання даних по даних форми, вивести результат

**Добавлення даних:** Введення даних, перевірка даних, якщо на дані дати немає резерву то створити його і добавити в табрицю результату в іншому випадку повідомити про помилку

**Зміни статусу резерву.** У таблиці резервів повинна бути передбачена можливість нажати на іконку зміни статусу з відправкою запиту на зміну статусу та зміни самої іконки на відповідну