II dalis. WEB servisų programavimo pagrindai

**Darbo tikslas**: Išmokti kurti sistemas naudojantis webservisais

Teorija

Duomenų apsikeitimui tarp programinių sistemų reikalingas vienodai interpretuojamas formatas. Klasikinis yra XML formatas, tačiau jis dažniau naudojamas dokumentų ar duomenų bazių struktūrai aprašyti. Keičiantis duomenimis tarp įvairių internetinių sistemų populiaresnis JSON formatas. Abiem atvejais tarp sistemų perduodami tik duomenys ir tik tekstiniu formatu (Unicode palaikomas). Jų atvaizdavimas ( formatai, spalvos, stiliai) neperduodamas. Tai kliento programos reikalas**.**

1. **JSON notacija.**

Duomenys JSON formatu atvaizduojami vardas:reikšmė poromis, atskirtomis kableliais Riestuose skliaustuose talpinami objektai, kampuotuose – masyvai. JSON identiškas JavaScript objekto formatui, tik tarp kabučių rašomi tiek vardai, tiek reikšmės. Pvz.:

*{“svečiai“:[*

*{“vardas“:“Jonas“, “pavardė“:“Gricius“},*

*{“vardas“:“Juras“, “pavardė“:“Gribius“}*

*]}*

**JavaSriptu** JSON tekstas konvertuojamas į objektą funkcija JSON.parse():

*let obj = JSON.parse(txt);*

Atvirkščias veiksmas: *let* *txt= JSON.stringify(obj);*

**PHP** atitinkamos funkcijos json\_decode() ir json\_encode():

*$txt = json\_encode($myArray);  
echo $txt;*

1. **Duomenų perdavimas JSON formatu tarp sistemų**

Ankstesniame lab. darbe padaryta monolitinė žinučių internetinė sistema: duomenų bazė ir su ja dirbanti programa tame pačiame serveryje, duomenų išvedimas ir vartotojo interfeisas fiksuotas. Tačiau sudėtingesniais atvejais naudojamas kitas metodas. Serverinė dalis (backend) valdo su DB susietas operacijas. Jų pasiekimui iš išorinių aplikacijų per internetą pateikiamas API. Tada dirbti su šia DB gali įvairios aplikacijos (frontend). Frontend gali veikti tiesiog vartotojo kompiuteryje, pvz. atidarant html failą su naršykle (veiks šio lab. d. 2 punkte). Tai suteikia naujas aplikacijų kūrimo galimybes. Šiame laboratoriniame darbe pagal duotus pavyzdžius reikės parašyti dvi programą naudojant jūsų jau sukurtą žinučių lentelę.

Serverinė (backend) dalis-PHP, nebendrauja su vartotoju, neturi įvedimo formų, neišveda į ekraną. Ji veikia Apache web serveryje, pagal gautą užklausą išrenka iš lentelės duomenis ir grąžina atsakymą JSON formate frontend daliai.

Klientinė (frontend) dalis - HTML su JavaScript. Skirtingai nei PHP, Javascript veikia naršyklėje. Vartotojui pateikia įvedimo formą, siunčia užklausą į backe nd ir išveda jo atsakymą. Užklausos iš

***Javascript***

Frontend programoje naršyklė pirma parodo iš serverio gautą tinklapio html, o po to Javascript jį papildo ar modifikuoja suteikdamas tinklapiui dinamiškumo. **JavaScript** visi tinklapio elementai sudėti tam tikra hierarchine tvarka. Kiekvienas elementas yra objektas. Kiekvienas objektas turi metodus ir savybes. Objektai suteikia galimybę susieti savybes su reikšmėmis. Lab. darbo pavyzdžio programoje naudojamas populiarus JavaScript karkasas **JQuery**. Iš klientinės dalies į serverinę dalį bus siunčiamos užklausos naudojant JQuery bibliotekos Ajax funkcijas

(plačiau: <https://www.w3schools.com/jquery/jquery_ajax_get_post.asp>).

Darbo eiga

1. **Sukurti lentelę pavadinimu: vardas\_pavarde\_lab**

Domenų bazę pasieksite adresu: <http://localhost/phpmyadmin>

* Prisijungimo vardas: stud, slaptažodis: stud

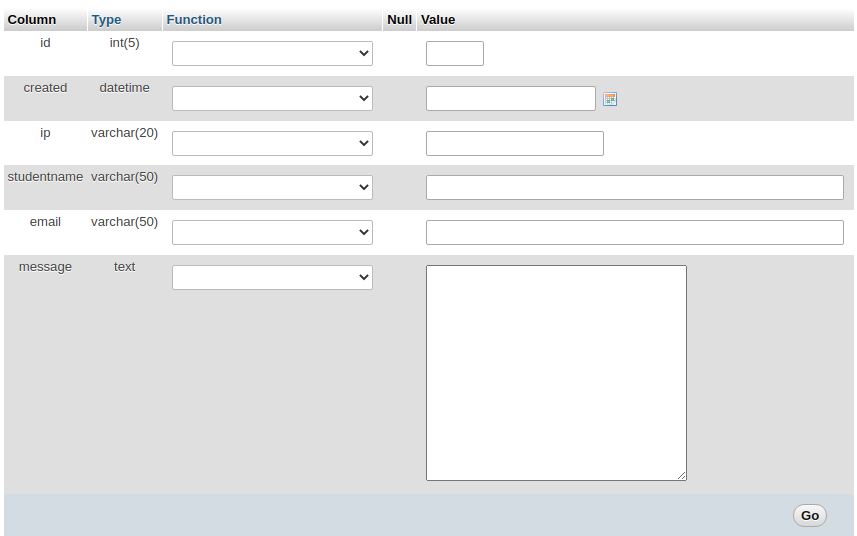
**vardas\_pavarde\_lab** lentelės laukai:

* **id**: INT(5), Primary, autoincrement;
* **created**: datetime;
* **ip**: VARCHAR(20);
* **studentname**: VARCHAR(50);
* **email**: VARCHAR(50);
* **message**: TEXT;

*Galite naudoti prieš tai buvusio laboratorinio darbo lentelę*.

1. **Užpildyti duomenų bazės lentelę testiniais duomenimis**

Duomenų užpildymo forma pasiekiama: (Meniu: Insert 1 pav.)



**1 pav**. Lentelės užpildymo forma

Pasitikrinimui ar duomenys išsisaugojo duomenų bazėje pažiūrėkite sąrašą (2 pav. meniu: Browse)

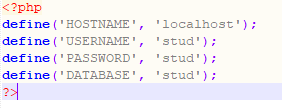


**2 pav.** Lentelės įrašų sąrašas

1. **Sukurti konfigūracijai config.php failą**

Susikurkite katalogą **/var/www/html/lab3** ir jame talpinkite sukurtus failus.

Sukūrę config.php failą išsaugokite, jame įrašykite konstantas prisijungimų prie duomenų bazės (3 pav.):



**3 pav.** config.php failo turinys

1. **Sukurti lentelės skaitymo programą read.php**

Ši programa (4 pav.) atsakinga už duomenų nuskaitymą iš duomenų bazės. Rezultatą programa turi grąžinti JSON formatu. Atidarius nuorodą <http://localhost/lab3/read.php> turėtumėte pamatyti JSON rezultatą.

|  |  |
| --- | --- |
|  |  |
|  |

**4 pav.** read.php failo turinys

Pastaba: Jei lentelės reikšmėse naudojamos lietuviškos raidės (Collation utf8\_lithuanian\_ci), prieš skaitymą pridėkite

**mysqli\_set\_charset($dbc,"utf8");**

1. **Sukurti vartotojo programą plepys.html ir ją patalpinti ant Desktop**

Kad veiktų Jūsų plepys.html aplikacija esanti ant Desktopo, o ne serveryje, reikia read.php programą papildyti antraštėmis pridedant:

header("Content-Type: application/json; charset=UTF-8");

header("Access-Control-Allow-Origin: \*"); *// Ši antraštė leis programai atsakyti į bet kieno užklausą*

**plepys.html** atsakingas už duomenų paėmimą iš read.php programos/webserviso ir jų atvaizdavimą. Šį failą papildykite eilutėmis:

*Į HEAD dalį įdėkite eilutę***:**

<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/css/bootstrap.min.css" integrity="sha384-GJzZqFGwb1QTTN6wy59ffF1BuGJpLSa9DkKMp0DgiMDm4iYMj70gZWKYbI706tWS" crossorigin="anonymous">

*BODY gairėje patalpinkite:*

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi6jizo" crossorigin="anonymous"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.6/dist/umd/popper.min.js" integrity="sha384-wHAiFfRlMFy6i5SRaxvfOCifBUQy1xHdJ/yoi7FRNXMRBu5WHdZYu1hA6ZOblgut" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.2.1/dist/js/bootstrap.min.js" integrity="sha384-B0UglyR+jN6CkvvICOB2joaf5I4l3gm9GU6Hc1og6Ls7i6U/mkkaduKaBhlAXv9k" crossorigin="anonymous"></script>

*Taip pat BODY srityje įdėkite DIV‘ą kuriame norėsite atvaizduoti rezultatus*lentelės formatu:

<div id="table"></div>

1. **Pridėti duomenų nuskaitymo kodą plepys.html** **faile**

Papildykite kodą JQUERY programiniu kodu, kuris iš **read.php** programos API nuskaitys duomenis ir juos atvaizduos lentelės pavidalu į <div id="table"></div> sritį:

<script>

$(document).ready(function () {

//nuskaitomos žinutės užkrovus puslapį

readMessages();

});

// funkcija atsakinga už žinučių nuskaitymą iš read API

function readMessages()

{

$.ajax({

url: 'http://localhost/lab3/read.php',

type: 'GET',

data: { },

success: function (data) {

// padarykite lentelę su tais pačiais stiliais kaip php.lab

// Pagal anstesnį lab: <table style="margin: 0px auto;" id="table"> ... </table>

var table = '<table class="table table-striped" table style="margin: 0px auto;" id="table">';

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

{

table = table + '<tr><td>' + data[i].studentname +…

…+ '</td></tr>';

}

table = table + '</table>';

$('#table').html(table);

}

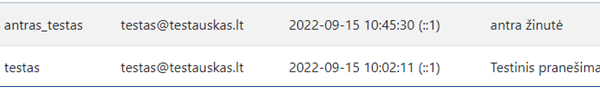
});

}

</script>

1. **Patestuoti duomenų atvaizdavimą plepys.html faile**

Patikrinkite naršyklėje ar duomenys jau atvaizduojami lentelės pavidalu, kaip pavaizduota 5 pav.



**5 pav.** Duomenų atvaizdavimas lentelės pavidalu

Jei atvaizdavimas sėkmingas, tuomet keliaujame prie duomenų įterpimo formos.

1. **Sukurti formą plepys.html faile vartotojo programoje**

Sukuriame formą, kaip pavaizduota 6 paveikslėlyje. Ji atitinka ankstesnio lab2 darbo stilius su šiam darbui reikalingais įvedimo laukais:

|  |
| --- |
|  |

**6 pav.** Duomenų formos sukūrimas

Parametro ACTION nenurodykite. Tokiu atveju programa siųs duomenis pati į save. Pasižiūrime ar forma vaizduojama. Ir keliaujame prie duomenų įrašymo **write.php** programos.

1. **Sukurti write.php programą programą**

Programoje (7 pav.) patikrinama ar yra siunčiami POST metodu duomenys, jei taip, jungiamasi prie duomenų bazės ir įterpiami į ją duomenys. Patestuoti galima **plepys.html** vartotojo programoje FORM tage pridėjus action=<http://localhost/lab3/write.php>. Duomenų bazėje turėtų atsirasti įrašai, tačiau patestavus, jei veikia, ištrinkite šį action, nes duomenis perdavinėsime AJAX metodu.



**7 pav.** write.php failo turinys

Jei failas veikia, važiuojam toliau!

**write.php** programą papildykite antraštėmis, kurių reikia, kad ir atsakymas sėkmingai būtų gaunamas:

header("Content-Type: application/json; charset=UTF-8");

header("Access-Control-Allow-Origin: \*");

1. **Pamodifikuoti plepys.html vartotojo programą**

$(document).ready() funkcijoje pridėkite skriptą, kuris iš Jūsų formos duomenis perduos į write.php programą ir po sėkmingo duomenų perdavimo, atnaujins lentelės duomenis:

// paspaudus siųsti ant formos, siunčiami duomenys į serverį

$("#forma").submit(function() {

var datastring = $("#forma").serialize(); // formos duomenys paruošiami siuntimui

$.ajax({

url: 'http://localhost/lab3/write.php',

type: 'POST',

data: datastring,

success: function (data) {

// sėkmės atveju iš naujo nuskaitoma lentelė

readMessages();

}

});

return false;

})

Po šio kodo pridėjimo, žinučių siuntimo forma turėtų pilnai veikti.