МІНІСТЕРСТВО ОСВІТИ I НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ   
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

ФАКУЛЬТЕТ БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ

КАФЕДРА БІОМЕДИЧНОЇ КІБЕРНЕТИКИ

**Практична робота №4**

з дисципліни «Розробка крос-платформного програмного забезпечення»

тема: Багатоланкова архітектура. «Front-end» та «back-end».

Товсті та тонкі клієнти. Насичені інтернет-застосунки. AJAX

**Виконав:**

Студент гр. БС-03

Затуловський Г. А.

**Перевірела:**

ст.викл. Сердаковський В. С.

ac. Рисін. С. В.

Зараховано від \_\_\_.\_\_\_.\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис викладача)

Київ-2023

**Мета:** ознайомитися з поняттям та видами масштабованості; виявити роль інтерфейсу в складових системах; з'ясувати різницю між архітектурами програмних систем, сильні та слабки сторони кожної з них; визначити переваги та недоліки товстих та тонких клієнтів; порівняти традиційні та «насичені» інтернет-застосунки, ознайомитися з технологіями, що можуть бути застосовані для реалізації останніх; вивчити принципи побудови та використання підходу AJAX; на основі шаблону AJAX організувати роботу з колекцією об'єктів, що описані у варіанті завдання.

## **Завдання:**

1. Код попередньої роботи переналаштувати на з'єднання з зовнішнім сервером та організувати можливість працювати у програмі з віддаленою базою даних.

2. Завантажити з сервера шаблони файлів.

3. Відповідно до своєго варіанту встановити значення для групи, варіанту, автора та паролю в файлі «user.php»

4. Додати код для формування на вихідній сторінці вірних значень «IP-адреса», «дата та час», «час формування сторінки» в файлі «get.php».

5. Відповідно до опису сутності у варіанті :

5.1. сформувати опис полів та тексти запитів до бази даних в файлі «db.php»;

5.2. додати код для перевірки значень полів на обмеження та іх зберігання в базі даних в файлі «post.php»;

5.3. додати код для відображення полів в файлі «get.php»;

5.4. сформувати опис полів та додати код для перевірки їх значень на обмеження в файлі «get.js»;

5.5. додати стилі для підсвітки вибраного рядка таблиці та рядка під курсором та відображення курсору «вказівник» над клітинками таблиці у файлі «get.css».

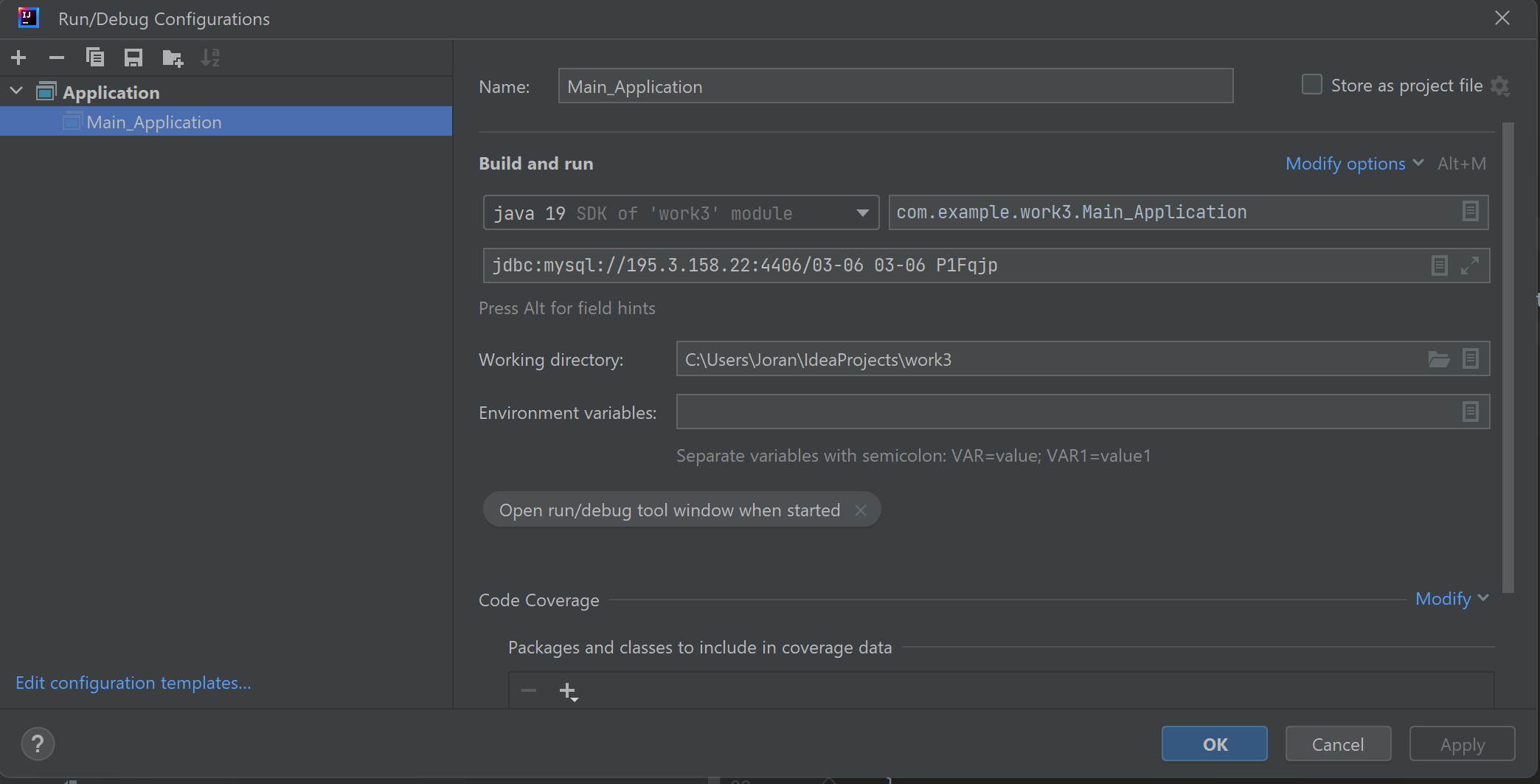
6. За допомогою ftp-клієнта завантажити дороблені файли до особистого каталогу сервера.

## Варіанти:

| **Номер варіанту** | **Об'єкт** |
| --- | --- |
| 6. | Вистава: назва п'єси, автор, театр, дата вистави, кількість актів, ціна квитка, музикальний супровід (так/ні) |

## **Лістинг програми:**

Зміна в java:

db.php

<?php  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
//  
// fields : db name, html name, html title, maxlen, editable  
  
// id field  
define('\_id\_', 'id');  
  
// table name  
define('\_tbl\_name\_', 'data');  
  
// field metadata indices  
define('\_fld\_name\_', *0*);  
define('\_fld\_fld\_', *1*);  
define('\_fld\_head\_', *2*);  
define('\_fld\_len\_', *3*);  
define('\_fld\_edit\_', *4*);  
  
// field metadata  
$fields  
 = array  
 (  
 array ( *\_id\_*, *\_id\_* , 'Id' , *0*, *false* ),  
 array ( 'name', 'nm', 'Name', *100*, *true* ),  
 array ( 'author', 'aut' , 'Author' , *100*, *true* ),  
 array ( 'theatre', 'the', 'Theatre' , *100*, *true* ),  
 array ( 'performance', 'per', 'Perfomance' , *100*, *true* ),  
 array ( 'quantity\_of\_acts', 'quan', 'Quantity of acts' , *100*, *true* ),  
 array ( 'ticket\_price', 'tick', 'Ticket price' , *100*, *true* ),  
 array ( 'music', 'mus', 'Music' , *1*, *true* )  
 );  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
//  
// database queries  
  
define('\_tbls\_show\_', 'show tables');  
  
define('\_tbl\_make\_', '  
create table `' . *\_tbl\_name\_* . '`  
(  
 `id` bigint(20) not null,  
 `name` varchar(100) not null,  
 `author` varchar(100) not null,  
 `theatre` varchar(100) not null,  
 `performance` varchar(100) not null,  
 `quantity\_of\_acts` smallint not null,  
 `ticket\_price` bigint not null,  
 `music` boolean not null,  
 primary key (`id`)  
)  
engine=myisam  
default charset=utf8;  
');  
  
define('\_tbl\_drop\_', 'drop table `' . *\_tbl\_name\_* . '`');  
  
define('\_tbl\_wipe\_', 'delete from `' . *\_tbl\_name\_* . '`');  
  
define('\_tbl\_seed\_', "  
insert into `" . *\_tbl\_name\_* . "` (`" . $fields[*0*][*\_fld\_name\_*] . "`, `" . $fields[*1*][*\_fld\_name\_*] . "`, `" . $fields[*2*][*\_fld\_name\_*] . "`, `" . $fields[*3*][*\_fld\_name\_*] . "`, `" . $fields[*4*][*\_fld\_name\_*] . "`, `" . $fields[*5*][*\_fld\_name\_*] . "`, `" . $fields[*6*][*\_fld\_name\_*] . "`, `" . $fields[*7*][*\_fld\_name\_*] . "`) values  
(1, 'SORCERY\_POTION', 'Oleksandr\_Melnyk', 'Theater\_on\_Podil','3/12/2022','4','200','1' ),  
(2, 'SNOW\_IN\_APRIL', 'Taras\_Kravchenko', 'Theater\_19','10/10/2023','4','150','1'),  
(3, 'OVERCOAT', 'Maksym\_Oliynyk', 'Theater\_of\_Taras','2/6/2023','2','125','0'),  
(4, 'DREAMS\_COME\_TO\_LIFE', 'George\_Bondar', 'Academic\_Theater\_of\_Les\_Kurbas','2/16/2022','6','375','0'),  
(5, 'OBSERVER', 'Vasyl\_Rudenko', 'Theater\_on\_Chinay\_19','24/12/2023','5','230','1');  
");  
  
define('\_tbl\_insrepl\_', ' into `' . *\_tbl\_name\_* . '` (`id`, `name`, `author`, `theatre`, `performance`, `quantity\_of\_acts`,`ticket\_price`,`music`) values (?, ?, ?, ?, ?, ?, ?, ?)');  
  
define('\_tbl\_insert\_', 'insert' . *\_tbl\_insrepl\_*);  
  
define('\_tbl\_replace\_', 'replace' . *\_tbl\_insrepl\_*);  
  
define('\_tbl\_delete\_', 'delete from `' . *\_tbl\_name\_* . '` where `id` = ?');  
  
define('\_tbl\_select\_', 'select \* from `' . *\_tbl\_name\_* . '` order by `id`');  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
//  
// returns mysqli or stmt error  
  
function show\_error ( $mysqli, $stmt = *null* )  
{  
 if ( ! $stmt )  
 if ( $mysqli->connect\_errno )  
 return 'error ' . $mysqli->connect\_errno . ' connecting to db :' . '<br>' . $mysqli->connect\_error;  
 else  
 return 'error ' . $mysqli->errno . ' executing query :' . '<br>' . $mysqli->error;  
 else  
 return 'error ' . $stmt->errno . ' executing query :' . '<br>' . $stmt->error;  
}  
  
//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
  
?>

get.php

$start=microtime();  
date\_default\_timezone\_set('Europe/Kiev');

<?php  
  
 $ok  
 = $stmt->execute();  
 if ( $ok )  
 // selecting data  
 {  
  
 $vals  
 = array();  
 $stmt->bind\_result  
 (  
 $vals[$fields[*0*][*\_fld\_fld\_*]], $vals[$fields[*1*][*\_fld\_fld\_*]], $vals[$fields[*2*][*\_fld\_fld\_*]],$vals[$fields[*3*][*\_fld\_fld\_*]],  
 $vals[$fields[*4*][*\_fld\_fld\_*]], $vals[$fields[*5*][*\_fld\_fld\_*]], $vals[$fields[*6*][*\_fld\_fld\_*]],$vals[$fields[*7*][*\_fld\_fld\_*]]  
 );  
  
 for ( $max\_id = *0*; $stmt->fetch(); )  
 // walking through records  
 {  
 if ( $vals[$fields[*0*][*\_fld\_fld\_*]] > $max\_id )  
 $max\_id = $vals[$fields[*0*][*\_fld\_fld\_*]];  
  
?>

Группа БС-<?=*\_grp\_*?>, вариант <?=*\_var\_*?>, <?=*\_authname\_*?>  
<hr>  
Ваш IP-адрес : <?=$\_SERVER['REMOTE\_ADDR']?>, текущие дата и время : <?=date('m/d/Y h:i:s a', time())?>, страница сформирована за : <?=microtime()-$start?> сек  
<hr>

get.js

// fields ids  
let  
 fields = [ \_id\_, 'nm', 'aut', 'the', 'per', 'quan' , 'tick', 'mus' ];

function change\_item ( anEvt )  
{  
let  
 elem = anEvt.target;  
  
 if ( elem.tagName != 'INPUT' )  
 return false;  
  
 // checking field for emptiness  
let  
 ok = elem.value.trim().length;  
  
 if ( ok )  
 // checking field values  
 switch ( elem.name )  
 {  
  
 case 'quan' : let quantity\_of\_acts = parseInt(elem.value); ok = quantity\_of\_acts > 0; break;  
 case 'tick' : let ticket\_price = parseInt(elem.value); ok = ticket\_price > 0; break;  
 case 'mus' : let mus = parseInt(elem.value); ok = mus === 0 || mus === 1; break;  
 }  
  
 if ( elem.value.trim() == elem.title )  
 elem.style.removeProperty('background-color');  
 else  
 elem.style.backgroundColor = ok ? '#FF7' : '#FBB';  
  
 return true;  
}

get.css

body  
{  
 font-family : Tahoma, Verdana, Sans;  
 background : #BBF;  
 margin : 1em;  
}  
  
th  
{  
 border-bottom : 1px blue solid;  
 border-right : 1px blue solid;  
 padding : 0 0.5em;  
}  
  
table  
{  
 border-collapse : collapse;  
 border-radius : 2px;  
}  
  
table td  
{  
 padding : 0;  
}  
  
table#list  
{  
 margin : 1em 0;  
}  
  
table#list, table#dtls  
{  
 border : 2px blue solid;  
}  
  
table#list tr#id\_0  
{  
 display : none;  
 background-color : #79F;  
}  
  
table#list tr#id\_0.show  
{  
 display : table-row;  
}  
  
table#list td, table#dtls td  
{  
 padding : 0.125em 0.5em;  
}  
  
table#list td, table#list th  
{  
 border-right : 1px blue solid;  
}  
  
table#list td:first-child  
{  
 text-align : right;  
}  
  
table#list td input  
{  
 padding : 0;  
 margin-right : 0.25em;  
 max-width : 1.5em;  
}  
  
table#dtls td  
{  
 border-bottom : 1px blue solid;  
}  
  
table#dtls td:first-child  
{  
 font-weight : bold;  
}  
  
table.ctrl, table.load  
{  
 margin-left : 1em;  
}  
  
table.ctrl td, table.load td  
{  
 padding : 0.5em 0;  
}  
  
table.ctrl input[type='button']  
{  
 width : 8em;  
}  
  
table.load input[type='button']  
{  
 width : 10em;  
}  
  
input[type='button']  
{  
 width : 10em;  
}  
  
td input[type='text']  
{  
 min-width : 20em;  
}  
  
td input[readonly]  
{  
 background : unset;  
}  
  
td input[name='id']  
{  
 text-align : right;  
}  
  
input[type='button']:not([disabled]):hover  
{  
 font-weight : bold;  
}  
  
input[name='make'], input[name='seed']  
{  
 margin-top : 0.5em;  
 color : blue;  
}  
  
input[name='wipe'], input[name='drop']  
{  
 margin-top : 1em;  
 color : red;  
}  
  
#log.show  
{  
 display : inline-block;  
 margin-top : 1em;  
 padding : 0.5em;  
 border : 2px red solid;  
 color : red;  
}

post.php

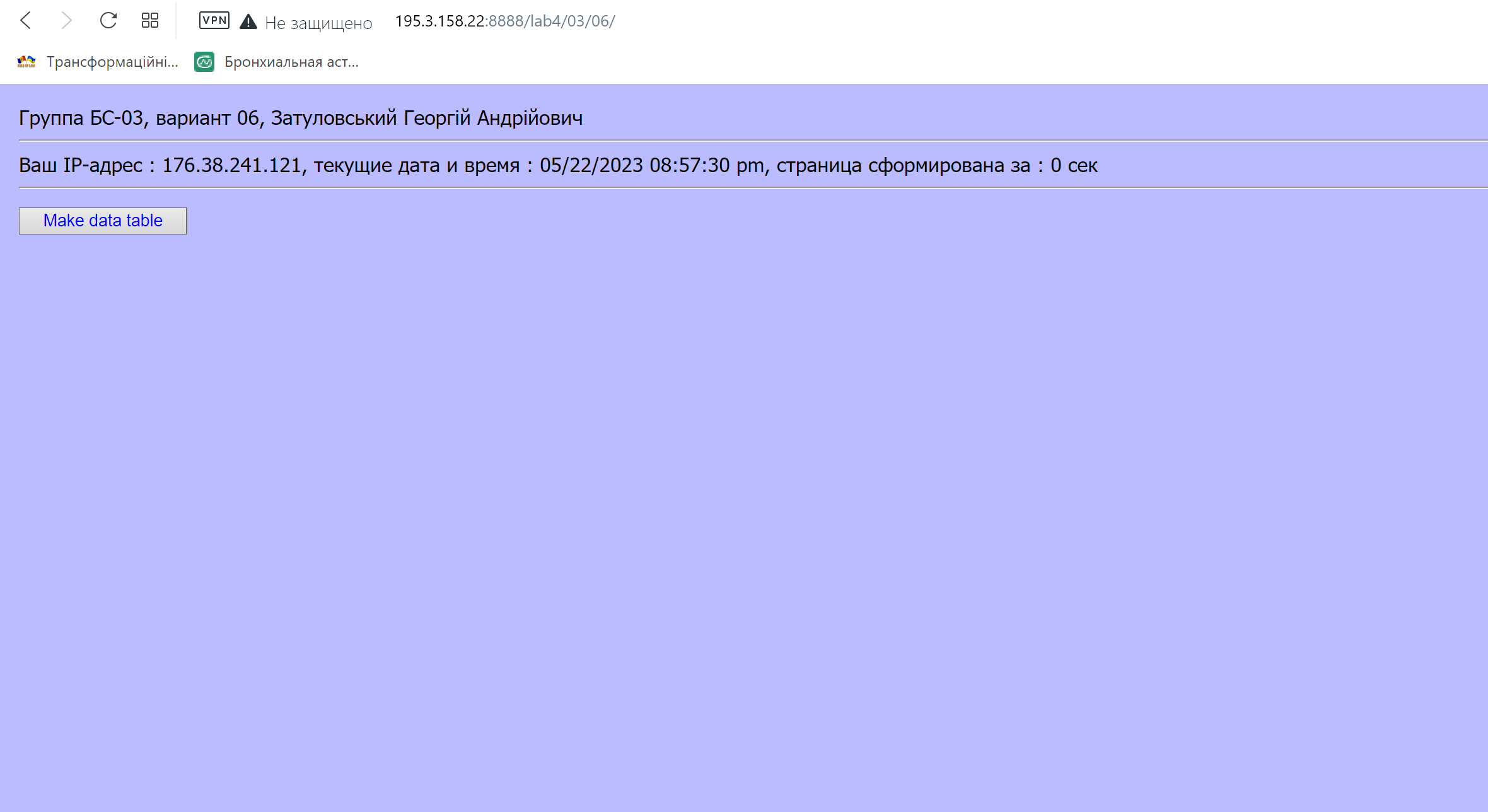
function check\_field ( $data )  
{  
$err  
 = *null*;  
$fldNo  
 = *0*;  
 // checking field for emptiness  
 foreach ( $data as $fld => $val )  
 if ( $fldNo++ && $val == '' )  
 $err .= 'error : field ' . "'" . $fld . "'" . ' mustn*\'*t be empty' . '<br>';  
  
 // checking field for righteousness  
 $quan = sscanf($data['quan'], '%d')[*0*];  
 $tick = sscanf($data['tick'], '%d')[*0*];  
 if ( ! ($quan > *0*) )  
 $err .= 'wrong field ' . "'quan'" . ' : the value less then zero : ' . "'" . $data['quan'] . "'" . '<br>';  
  
 if ( ! ($tick > *0*) )  
 $err .= 'wrong field ' . "'tick'" . ' : the value less then zero : ' . "'" . $data['tick'] . "'" . '<br>';  
 $mus = sscanf($data['mus'], '%d')[*0*];  
 if ( ! ($mus == *0* || $mus == *1*) )  
 $err .= 'wrong field ' . "'mus'" . ' : ' . 'the value must be zero or one' . ' : ' . "'" . $data['isdbt'] . "'" . '<br>';  
  
 return $err;  
}

function save\_item ( $mysqli, $data )  
{  
 // checking field values  
 if ( $err = check\_field($data) )  
 return 'error(s) :' . '<br>' . $err;  
  
$errs  
 ='';  
  
 // saving record  
 if ( $stmt = $mysqli->prepare(*\_tbl\_replace\_*) )  
 {  
 $stmt->bind\_param('dssssddd', $data[*\_id\_*], $data['nm'], $data['aut'], $data['the'], $data['per'], $data['quan'], $data['tick'], $data['mus']);  
  
 if ( $stmt->execute() )  
 {  
 if ( ! $stmt->affected\_rows )  
 $errs = 'error : no rows affected';  
 }  
 else  
 $errs = show\_error($mysqli, $stmt);  
  
 $stmt->close();  
 }  
 else  
 $errs = show\_error($mysqli);  
  
 return $errs;  
}

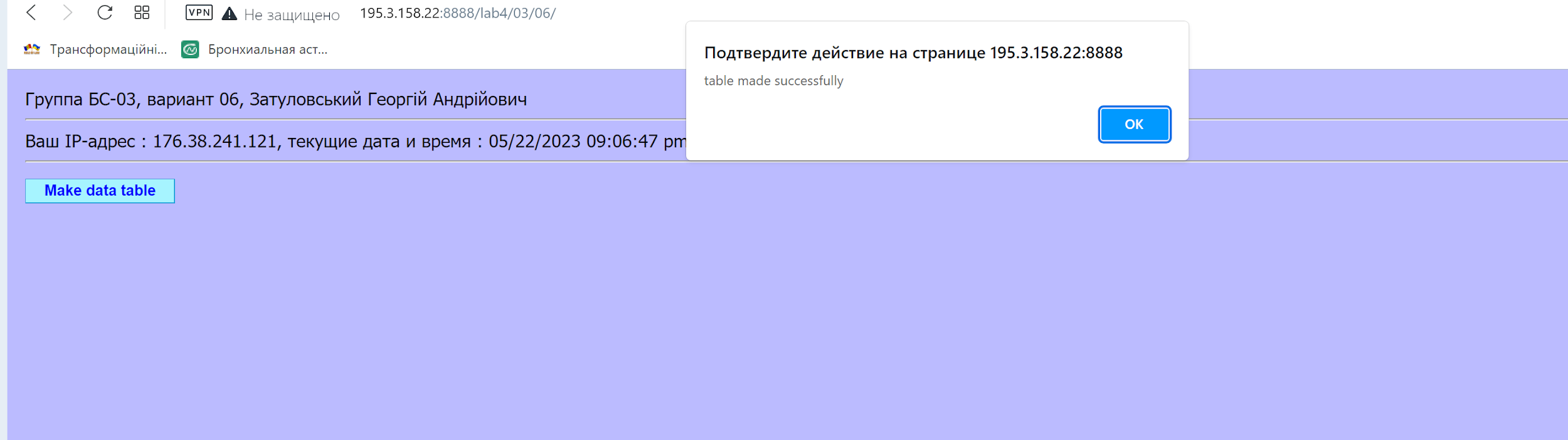
function load\_list ( $mysqli, $fields )  
{  
 if ( ! (count($\_FILES) && isset($\_FILES['csvi'])) )  
 // no file  
 {  
 echo 'error ' . 'no csv file';  
 return;  
 }  
  
$file  
 = $\_FILES['csvi'];  
 if ( $file['error'] != *0* )  
 // error uploading file  
 {  
 echo 'error ' . $file['error'] . ' uploading file';  
 return;  
 }  
 if ( ! move\_uploaded\_file($file['tmp\_name'], './' . $file['name']) )  
 // error moving uploaded file  
 {  
 echo 'error moving file';  
 return;  
 }  
  
$ids  
 = array();  
  
$errs  
 = '';  
  
 if ( ($hndl = fopen('./' . $file['name'], 'r')) !== *false* )  
 // processing file  
 {  
 $head  
 = trim(fgets($hndl));  
 $sep  
 = (strpos($head, "*\t*") !== *false* ? "*\t*" : ',');  
 $head = explode($sep, $head);  
 $idxs  
 = array();  
 // processing head row  
 foreach ( $fields as $fld )  
 if ( ($idx = array\_search($fld[*0*], $head)) !== *false* )  
 array\_push($idxs, $idx);  
 else  
 {  
 $errs .= 'error : ' . 'field ' . "'" . $fld[*0*] . "'" . ' not found in file';  
 break;  
 }  
  
 // processing data rows  
 for ( $rowNo = *0*; ! $errs && ($vals = fgetcsv($hndl, *1024*, $sep)) !== *false*; $rowNo++ )  
 {  
 $data  
 = array();  
 for ( $fldNo = *0*; $fldNo < count($vals); $fldNo++ )  
 $data[$fields[$fldNo][*1*]] = $vals[$idxs[$fldNo]];  
  
 // checking field values  
 if ( $err = check\_field($data) )  
 {  
 $errs .= 'line ' . ($rowNo + *1*) . ' : ' . $err;  
 break;  
 }  
  
 if ( $stmt = $mysqli->prepare(*\_tbl\_insert\_*) )  
 {  
 $stmt->bind\_param('dssssddd', $data[*\_id\_*], $data['nm'], $data['aut'], $data['the'], $data['per'], $data['quan'], $data['tick'], $data['mus']);  
  
 if ( $stmt->execute() )  
 {  
 if ( $stmt->affected\_rows )  
 array\_push($ids, array(*\_id\_* => $data[*\_id\_*]));  
 else  
 $errs .= 'no rows affected';  
 }  
 else  
 $errs .= show\_error($mysqli, $stmt);  
  
 $stmt->close();  
  
 if ( $errs )  
 break;  
 }  
 else  
 {  
 $errs .= show\_error($mysqli);  
 break;  
 }  
 }  
 fclose($hndl);  
 }  
  
 unlink('./' . $file['name']);  
  
 if ( $errs )  
 // deleting records  
 foreach ( $ids as $id )  
 del\_item($mysqli, $id);  
  
 echo $errs ? 'error(s) :' . '<br>' . $errs : 'ok';  
}

**Результати:**

Запуск програми:

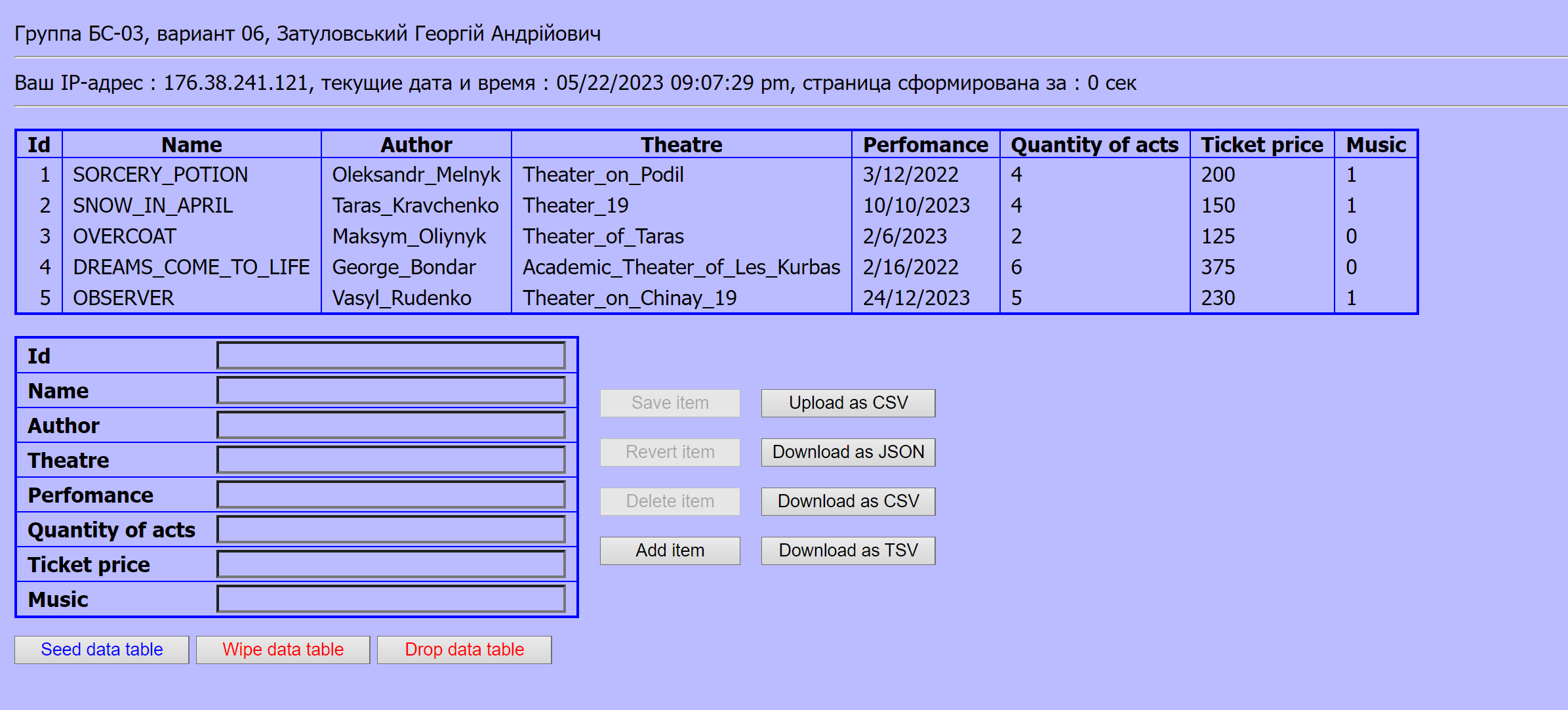


Створення таблиці:





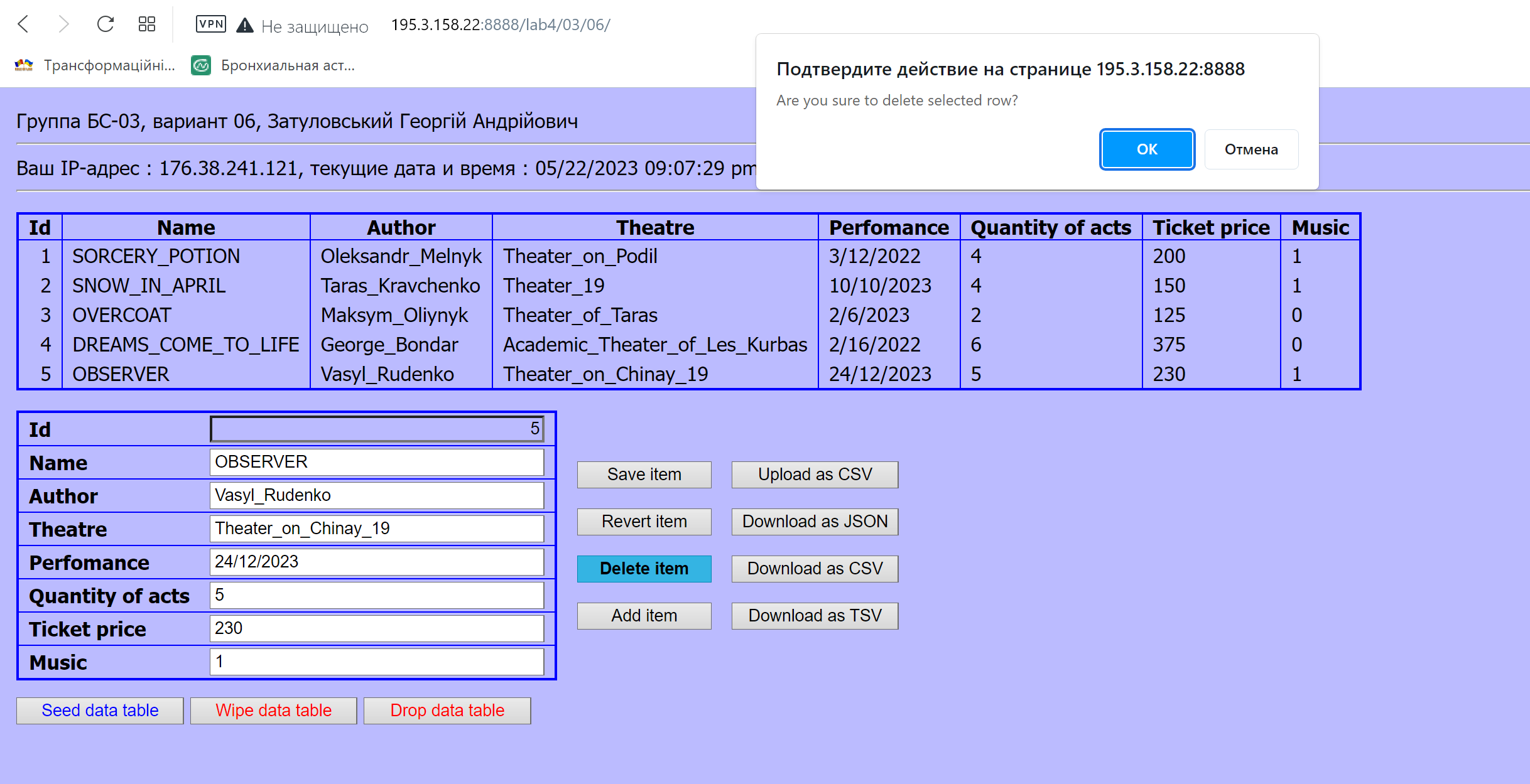
Завантаження даних:



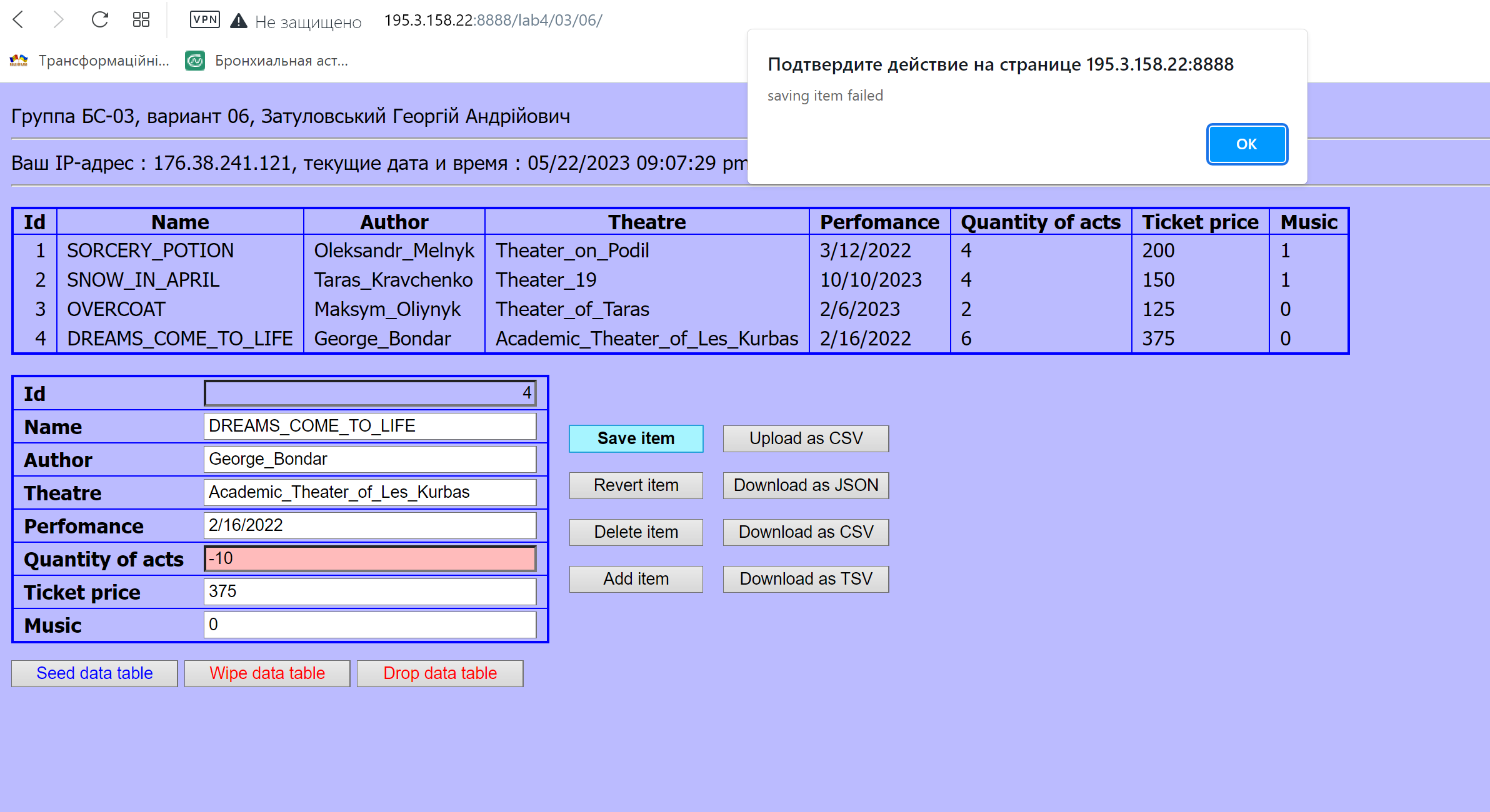
Обрання об’єкта:

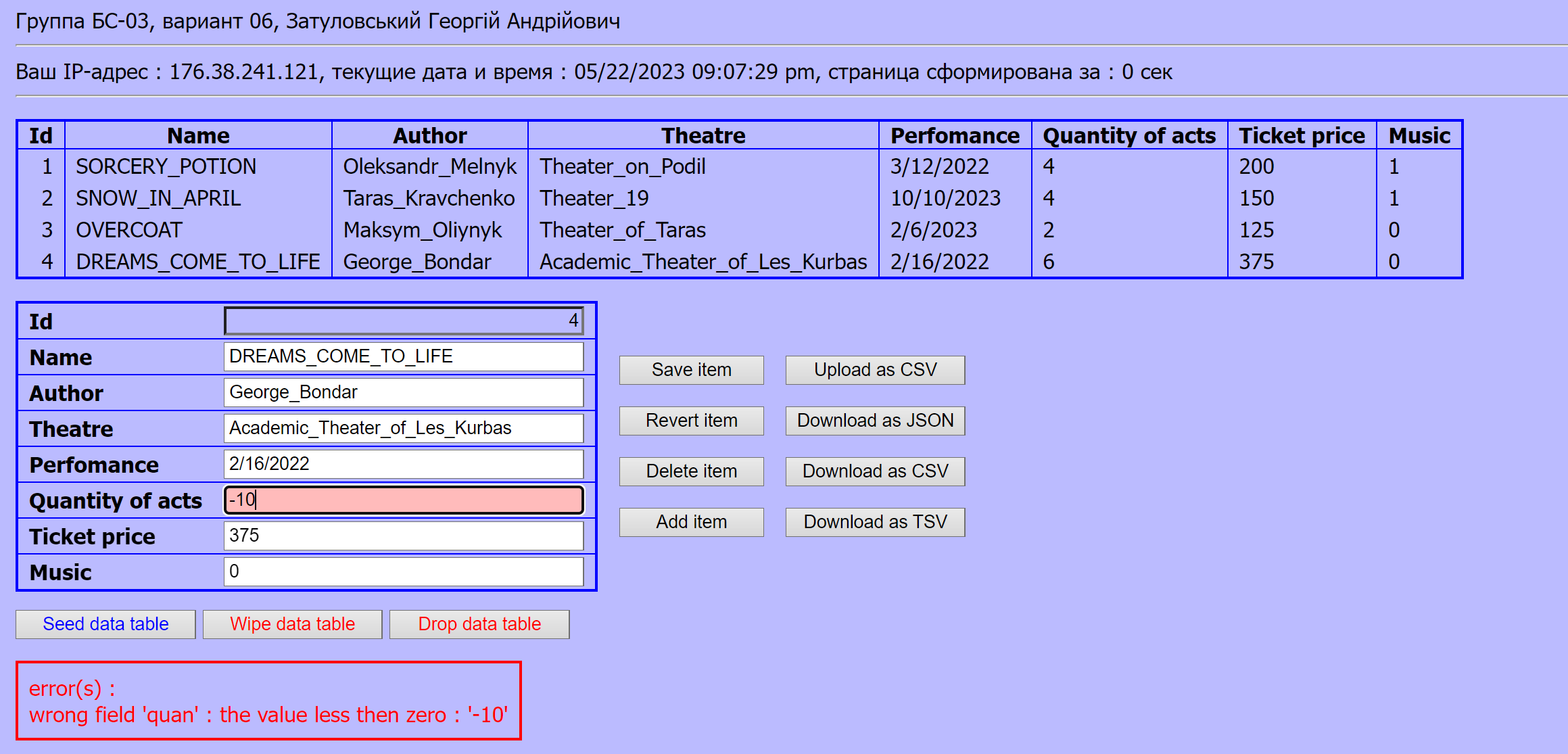


Видалення обраного об’єкту:

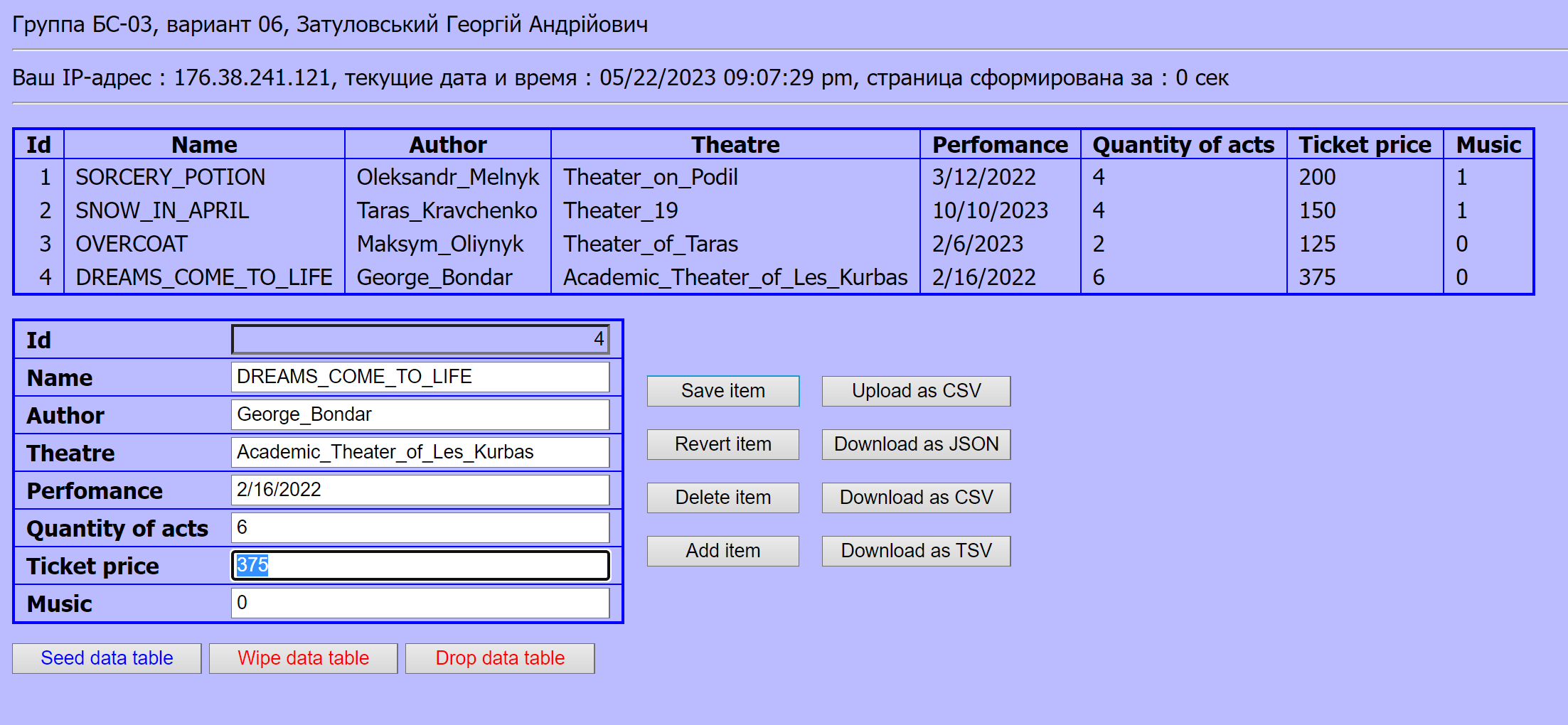


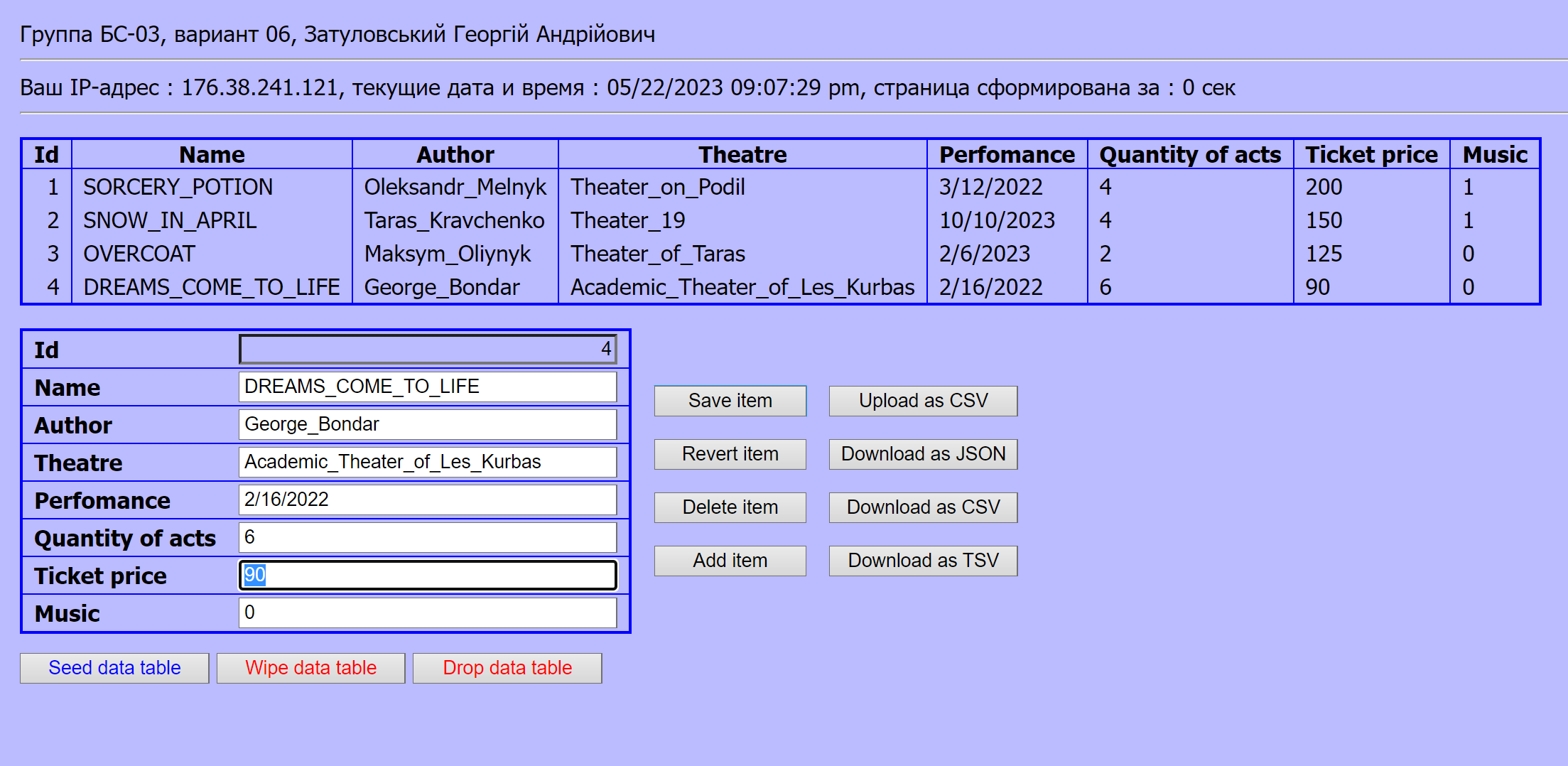
Невдала заміна об’єкту:





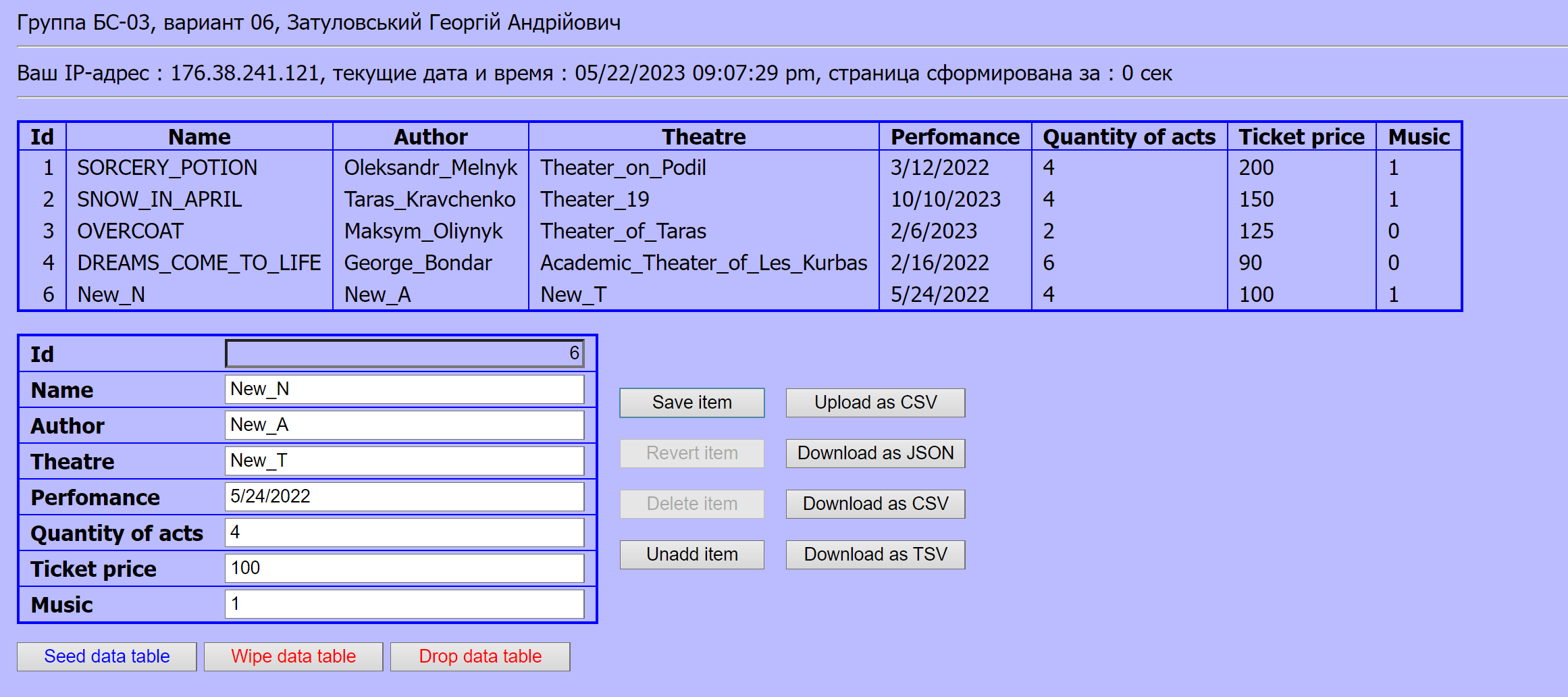
Вдала зміна об’єкту:



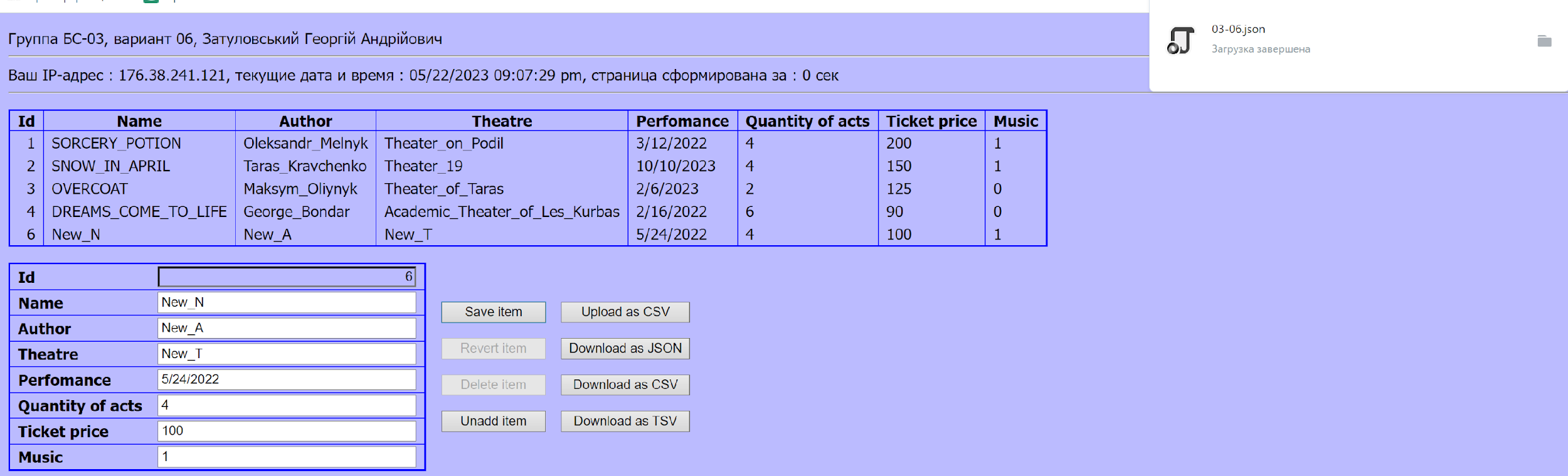


Додавання нового об’єкту:

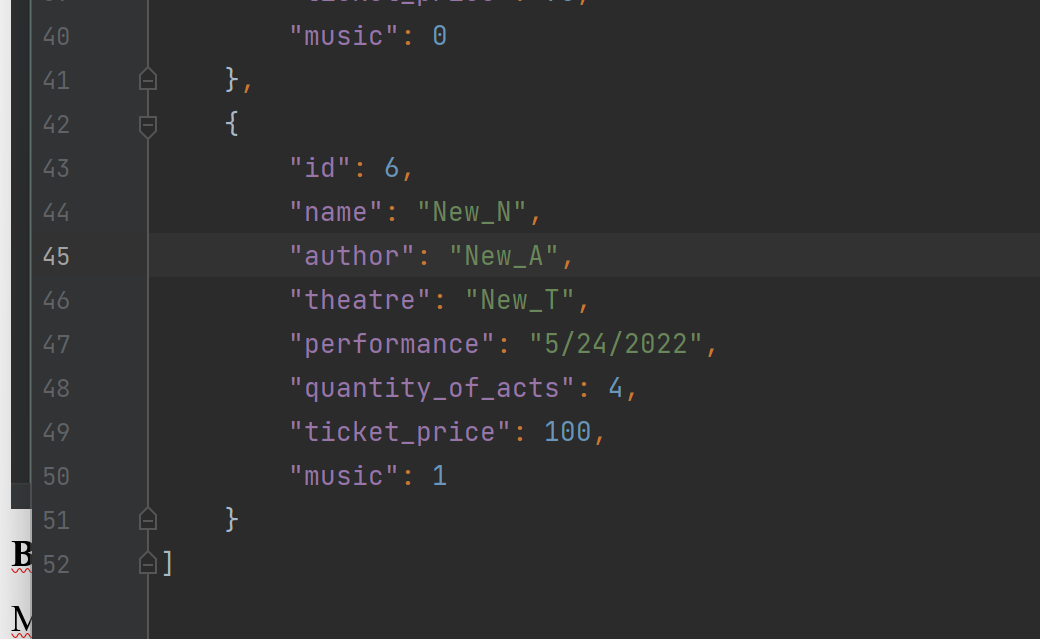
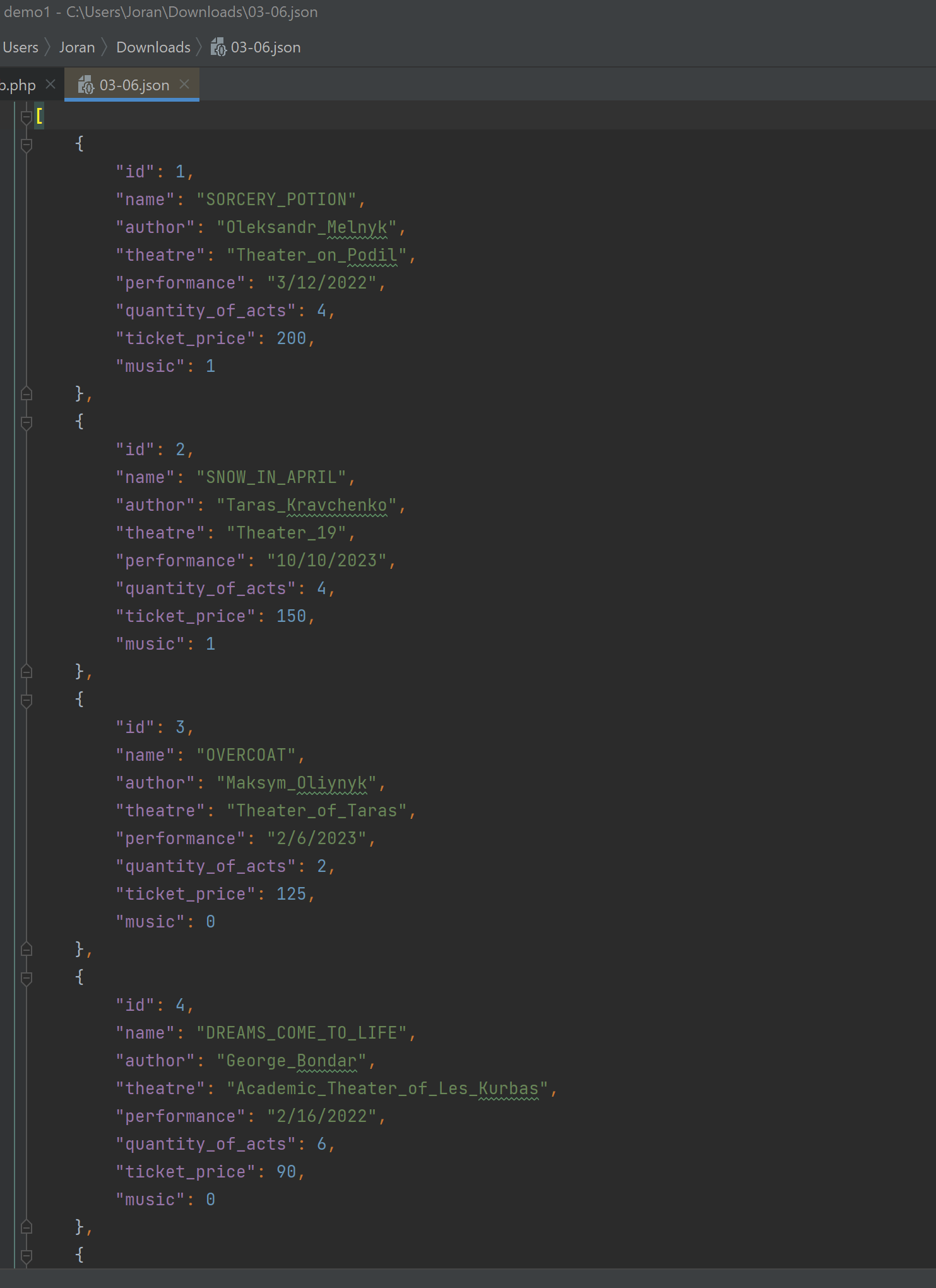




Збереження таблиці JSON:



Збережений файл JSON:

****

**Висновок:** ознайомилися з поняттям та видами масштабованості; виявити роль інтерфейсу в складових системах; з'ясували різницю між архітектурами програмних систем, сильні та слабки сторони кожної з них; визначити переваги та недоліки товстих та тонких клієнтів; порівняли традиційні та «насичені» інтернет-застосунки, ознайомитися з технологіями, що можуть бути застосовані для реалізації останніх; вивчити принципи побудови та використання підходу AJAX; на основі шаблону AJAX організували роботу з колекцією об'єктів, що описані у варіанті завдання.

## **Контрольні запитання:**

**1. Перелічіть основні види архітектур програмних систем та їх відмінності.**

1. Багаторівневий шаблон. принцип його полягає в тому, що вся система додатка розбивається на рівні, які відображаються на діаграмі. При цьому кожен рівень може викликати лише один інший рівень, що знаходиться нижче за нього. Це дає можливість вносити зміни до певних компонентів додатка, не зачіпаючи інші області. Але це ускладнює структуру архітектури та робить її досить навантаженою, що впливає на продуктивність.

2. Клієнт-серверна архітектура: У цій архітектурі компоненти системи розподілені між клієнтськими та серверними комп'ютерами. Клієнти взаємодіють з сервером, щоб отримувати доступ до ресурсів, таких як дані або послуги. Це дозволяє забезпечити більшу масштабованість та відновлюваність системи.

3. Ведучий-ведений - є одним із шаблонів архітектури програмного забезпечення, в якому визначається роль "ведучого" та "веденого" компонентів. Цей шаблон використовується для організації взаємодії між цими компонентами з метою досягнення ефективності та розділення завдань.

4. Канали і фільтри - цей шаблон підходить для систем, які виробляють і обробляють потоки даних. Кожен етап обробки відбувається всередині якогось компонента фільтра. Дані для обробки передаються через канали. Ці канали можна використовувати для буферизації або синхронізації даних.

**2. Відсутність якої складової відрізняє одноланкову архітектуру (як втілення монолітної структури) програмних систем від інших?**

Одноланкова архітектура програмних систем відрізняється від інших архітектур відсутністю розподіленої на менші компоненти.

**3. Опишіть розміщення та моделі взаємодії компонентів різних варіантів дволанкової архітектури та їх взаємні переваги і недоліки.**

Клієнт-серверну архітектуру можна означити, як концепцію інформаційної мережі в якій основна частина її ресурсів зосереджена в серверах, обслуговуючих своїх клієнтів. Така архітектура визначає такі типи компонентів:

* набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
* набір клієнтів, які використовують сервіси, що надаються серверами;
* мережа, яка забезпечує взаємодію між клієнтами та серверами.

Дволанкова клієнт-серверна архітектура передбачає взаємодію двох програмних модулів — клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

* модель тонкого клієнта, в рамках якої вся логіка застосунку та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
* модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта.

**4. В чому полягяють сильні та слабкі сторони інтернер-застосунків в контексті кросплатформності?**

Сильні сторони:

* Кросплатформовий доступ дає змогу користувачам використовувати їх на різних платформах, таких як комп'ютери, смартфони та планшети, незалежно від операційної системи.
* Оновлення веб-застосунків відбуваються на серверному боці, тому користувачам не потрібно завантажувати та встановлювати нові версії програмного забезпечення. Це полегшує розповсюдження оновлень та виправлень помилок.
* Кросплатформні інтернет-застосунки можуть мати єдиний дизайн та інтерфейс для всіх платформ, що полегшує їхнє використання для користувачів.

Слабкі сторони:

* Інтернет-застосунки мають обмежену можливість взаємодіяти з операційною системою пристрою. Не всі функції та можливості операційної системи можуть бути доступні для веб-застосунків.
* Інтернет-застосунки мають меншу швидкість роботи порівняно з нативними додатками, оскільки вони залежать від мережі та серверного зв'язку. Велика кількість запитів до сервера та повільне підключення можуть призвести до затримок у відображенні даних.

**5. Чи може сервер бути клієнтом? Поясніть на прикладі.**

Так, сервер може виступати клієнтом в розподілених системах або взаємодіяти з іншими серверами. Одним з прикладів може бути ситуація, коли сервер виконує запити до іншого сервера для отримання даних.

**6. Опишіть поняття товстого та тонкого клієнтів та їх взаємні переваги та недоліки.**

Товстий клієнт - це підхід, в якому більша частина обробки даних та виконання бізнес-логіки відбувається на стороні клієнта. У такій архітектурі клієнтська програма (наприклад, десктопний додаток) має значну кількість функцій та може працювати незалежно від сервера.

Переваги:

* Висока продуктивність
* Більша функціональність
* Наявність офлайн режиму

Недоліки:

* Складніше управління
* Більша вразливість до зломів
* Високі вимоги до обладнання

Тонкий клієнт - це підхід, в якому більша частина обробки даних та виконання бізнес-логіки здійснюється на стороні сервера. У такій архітектурі клієнтська програма (наприклад, веб-браузер) передає запити до сервера, а сервер здійснює обробку цих запитів та надсилає відповідь клієнту.

Переваги:

* Знижені витрати на обладнання
* Легше управління та підтримка
* Меншене навантаження на мережу

Недоліки:

* Залежність від мережі
* Обмежені можливості офлайн-режиму

**7. Розкрийте сутність концепції AJAX та вкажіть її плюси та мінуси порівняно з класичною.**

AJAX - це підхід до взаємодії з веб-сервером без перезавантаження сторінки. Він дозволяє асинхронно виконувати запити до сервера, отримувати дані та оновлювати вміст веб-сторінки без необхідності повного оновлення сторінки.

Плюси:

• Більша ресурсоємність на клієнтському пристрої

• Зменшення мережевого трафіку

• Покращена продуктивність

• Інтерактивність та динамічність

Мінуси:

• Збільшена складність розробки

• Проблеми з безпекою

• Залежність від JavaScript: AJAX

• Перевантаження сервера

**8. Які існують альтернативні AJAX технології побудови інтерактивних інтерфейсів?**

Приклади альтернативні AJAX технології побудови інтерактивних інтерфейсів: WebSocket, Server-Sent Events, Silverlight, JavaFX

**9. Чому додавання в обчислювальну систему додаткових ресурсів нерідко слабо або зовсім не підвищує продуктивність?**

Це відбувається тому що система може погано використовувати виділені ресурси, мати обмеження на використання ресурсів, мати погане масштабування, тому це може не підвищувати продуктивність

**10. На яких мовах реалізовані сервери інтернет-ресурсів, якими Ви користуєтесь?**

Python: Youtube, Instagram, Pinterest

C++: Google, Adobe Photoshop

Ruby: GitHub , Shopify

JavaScript: LinkedIn, Netflix, PayPal

PHP : Facebook, WordPress, Wikipedia.

Java: Amazon, Twitter, eBay.