

Ministerul Educației al Republicii Moldova

Universitatea Tehnică a Moldovei

Catedra Tehnologii Informaționale

RAPORT

Lucrarea de laborator#4

la Medii Interactive de Dezvoltare a Produselor Soft

A efectuat:

Profir Andrei

st.gr. TI – 143

A verificat:

Cojanu Irina

lect.asist.

Tema: Web Development

Scopul lucrării:

- Realizarea unui simplu Web Site personal
- Familiarizarea cu HTML si CSS
- Interactiuni Javascript

Formularea conditiei problemei:

- *Basic Level* (nota 5 || 6) :
 - Realizeaza un mini site cu 3 pagini statice
- *Normal Level* (nota 7 || 8):
 - Site-ul trebuie sa pastreze toata informatia intr-o baza de date
- *Advanced Level* (nota 9 || 10):
 - Site-ul trebuie sa contina AJAX Requests.
 - Implimentarea XHR sau JSON responses. Careva din informatie trebuie sa fie dinamic incarcata pe pagina.

Implementare task-uri:

- IDE: Zend Studio 13
- Web Server: Zend Server 8.5.3(PHPMyAdmin)
- Limbajul: PHP 5.6, JS, Sql
- Framework: Zend Framework 1.12.0
- Tehnologii: Pattern MVC

In acest laborator am folosit framework-ul Zend, acest framework insusi este bazat pe pattern-ul MVC, directoriile la crearea proiectului in Zend sunt aranjate si clasificate in forma MVC. Model, controller si view se afla in directorii aparte. Acest fapt ne permite ca sa creem proiecte foarte mari pe baza acestui framework.

In acest laborator a fost realizat un site care contine niste produse pentru a exemplifica unele tehnici de lucru cu limbajele ca JS, PHP si SQL.

Site-ul realizat contine 3 pagini, dintre care 2 pagini contin elemente care se incarca dinamic si o pagina statica. Pagina statica contine informatii privind contactele. Informatia de pe paginile dinamice se incarca din baza de date. In BD a fost creat un tabel cu denumirea products care contine mai multe cimpuri. Pe pagina principala avem un slider in care se incarca "random" citeva produse si avem posibilitatea sa facem refresh pentru a incarca in slider alte produse. Pentru a incarca alte produse a fost folosit AJAX ca pentru a renoi datele sa nu fie necesar de a face refresh la pagina. Ca rasuns la request este XHR dar datele cu care se lucreaza nu este XML dar este text obisnuit. Dar folosirea XHR ne permite sa utilizam XML dar aceasta nu este obligator.

```

<script type = "text/javascript">
    $(document).ready(refresh());
    function refresh() {
        var xhr = new XMLHttpRequest();
        xhr.open('GET', '/index/get-data-for-slider/', true);
        xhr.send();

        xhr.onreadystatechange = function() {
            if (xhr.readyState != 4) return;

            if (xhr.status != 200) {
                alert( xhr.status + ': ' + xhr.statusText );
            } else {
                $('<div id="cd-main-content">').html(xhr.responseText);
            }
        }
    }
</script>

```

Pe screen-ul dat putem vedea secventa de cod ce ne permite efectuarea unui AJAX request. "Get-data-for-slider" este un fisier view in care se afiseaza niste blocuri div ca ma apoi aceste blocuri sa formeze un slider pe care il putem observa pe pagina pincipala. Putem observa ca raspunsul primit cu ajutorul obiectului "xhr" este text fiindca sa apelat metoda "responseText" ce ne indica ca raspunsul primit de la request este un text simplu. Si in final putem atrage atentia ca a fost folosita metoda de transfer "GET" si nu "POST".

A doua pagina web care la fel se incarca dinamic este pagina "shop" dar datele in ea sunt incarcate cu simple "sql query" ceea ce inseamna ca datele din BD se primesc la fiecare incarcare a paginii sau la efectuarea unui refresh.

```

public function shopAction()
{
    // action body
    $productsModel = new Application_Model_DbTable_Products();
    $data = $productsModel->fetchAll()->toArray();
    $this->view->products = $data;
}

```

Aici este codul a action-ului "shop" in care se creeaza un nou obiect a model-ului ceea ce ne permite sa avem accesul la tabelul din BD in cazul nostru este tabelul cu denumirea "products".

Dupa ce se creeaza obiectul se face insusi extragerea din BD pentru aceasta simplu a fost folosta funtia "fetchAll()", ea permite sa extragem toate inregistrarile din tabelul la care ne arata "\$productsModel". In final datele primite cu ajutorul extragerii sunt transmise in view file. Acolo ele vor fi prelucrate si afisate. Afisarea datelor se poate efectua si in Controller in action-ul in care au fost extrase dar aceasta incalca regulile pattern-ului MVC.

Concluzie:

In aceasta lucrare de laborator a fost create un site care ne familiarizeaza cu HTML, CSS, JS, PHP. A fost elaborat un site care contine atat informatie statica cit si informatie dincamic incarcata din BD.

De asemenea s-a interactionat cu JS si anume AJAX ceea ce ne permite sa efectuam request-uri si sa afisam raspunsul la aceste request-uri fara a face refresh la pagina.