# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**ЗВІТ**

**Про виконання лабораторної роботи №3**

**«Опрацювання мовою Java Script**

**елементів форм HTML зворотнього зв'язку»**

**з дисципліни «Програмування інтернет застосувань»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняв:

ст. викл.Сенів М.М.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2014 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

ЛЬВІВ – 2014

**Тема роботи**: Опрацювання мовою Java Script елементів форм HTML зворотнього зв'язку.

**Мета роботи:** Оволодіти засобами керування формами, внесення інформації для сервера на динамічній web-сторінці.

## 1. Теоретичні відомості

HTML formФорма HTML містить групу елементів для приймання від користувача та пересилання даних у рядку запиту на Web-сервер. Ними є buttons, checkboxes, text fields, input etc.).

У тегі <form> є кілька нових атрибутів. Був доданий атрибут id, щоб не тільки семантично назвати цю форму, але також щоб надати унікальний ідентифікатор ID для ідентифікації форми, щоб її було легше оформити стилістично з допомогою CSS або, якщо буде потрібно, виконати маніпуляції за допомогою JavaScript. Можна мати тільки один унікальний id на сторінці, в цьому випадку ми отримуємо так звану contact-form.

Атрибут method визначає, як дані посилаються сценарієм, який буде їх обробляти. Двома найбільш поширеними методами є “GET” та “POST”. Метод “GET” посилає дані в URL браузера. Якщо немає спеціальної причини для використання “GET”, ймовірно, краще не використовувати його, якщо потрібно послати секретну інформацію, так як хто завгодно зможе побачити інформацію, що пересилається через URL. Метод “POST” посилає дані за допомогою сценарію, який підтримує форму, або в повідомленні e-mail, яке надсилається адміністратору сайту, або в базу даних для зберігання і подальшого використання, а не в URL, як метод “GET”. Метод “POST” є більш захищеним і зазвичай кращим варіантом.

Якщо ви стурбовані безпекою даних у формі, наприклад, якщо ви посилаєте номер кредитної карти на сайт магазину, то потрібно використовувати протокол https за допомогою рівня захищених сокетів (SSL). По суті це означає, що дані будуть надсилатися через протокол https, а не через протокол http.

Важливий атрибут action вказує на URL сервера та програму на ньому для опрацювання даних запиту. Атрибут method вказує на вибір метода кодування даних : get чи post. Наведемо приклад тегу форми,

<form action="http://www.facerec.com/app.php" method="get">

<fieldset>

<label>Name: <input type="text" name="name" /></label>

<label>sex: <input type="text" name="sex" /></label>

<label>country?

<input type="checkbox" name="country" />

</label>

<input id="sbutton" type="submit" value="Send me your picture!"/>

<fieldset>

</form>

## 2. Хід роботи

1. Побудувати веб-сторінку на вибрану тематику з полями на формі для введення назви прикріпленого файлу та перевірки його розширення. Текст файлу наведено нижче, результат роботи у браузері зображено на рис. 2.1.

<html>

<head>

<title>Лабораторна робота №3</title>

<script language = "JavaScript">

function validateForm() {

var x=document.forms["form1"]["surname"].value;

if (x==null || x=="") {

alert("Потрібно заповнити прізвище !");

return false;

}

x=document.forms["form1"]["resume"].value;

if(x.toLowerCase().indexOf(".docx") < 0) {

alert("Неправильне розширення файла !");

return false;

}

return true;

}

</script>

</head>

<body>

<form name = "form1" method="post" action="demo\_form.asp" onsubmit="return validateForm()">

Анкета претендента на вакансію:<br>

Прізвище: <input type = "text" name = "surname"><br>

Ім'я: <input type = "text" name = "name"><br>

Резюме (\*.docx)<input type="file" name="resume" id="resume">

<hr>

<input type = "submit" name = "submitBtn">

<input type = "reset" name = "resetBtn">

</form>

</body>

</html>

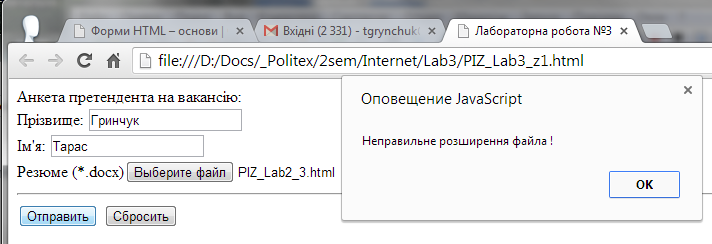


Рис. 2.1. Вигляд Web-документа завдання №1

у браузері Google Chrome

1. Побудувати веб-сторінку на вибрану тематику з полями на формі для введення назви програми для виконання на сервері, вигляд у браузері – рис. 2.2. Текст:

<html>

<head>

<title>Лабораторна робота №3</title>

<script language = "JavaScript">

function validateForm() {

var x=document.forms["form1"]["programName"].value;

if (x==null || x=="") {

alert("Потрібно заповнити назву сторінки для виконання!");

return false;

}

if(x.toLowerCase().indexOf(".asp") < 0) {

alert("Неправильне розширення файла !");

return false;

}

document.forms["form1"].action = x;

return true;

}

</script>

</head>

<body>

<form name = "form1" method="post" onsubmit="return validateForm()">

Запустити програму на сервері:<br>

Назва сторінки (\*.asp): <input type = "text" name = "programName"><br>

<hr>

<input type = "submit" name = "submitBtn">

<input type = "reset" name = "resetBtn">

</form>

</body>

</html>

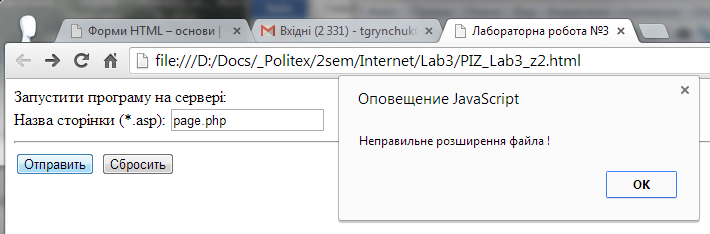
****

Рис. 2.2. Вигляд Web-документа завдання №2

у браузері Google Chrome

1. Побудувати веб-сторінку на вибрану тематику з полями на формі для вибору та введення назви методу передачі параметрів на сервер, вигляд у браузері – рис. 2.3. Текст:

<head>

<title>Лабораторна робота №3</title>

<script language = "JavaScript">

function validateForm() {

var x=document.forms["form1"]["surname"].value;

if (x==null || x=="") {

alert("Потрібно заповнити прізвище !");

return false;

}

if(document.forms["form1"]["meth1"].checked)

document.forms["form1"].method = "get";

else

document.forms["form1"].method = "post";

return true;

}

</script>

</head>

<body>

<form name = "form1" action="demo\_form.asp" onsubmit="return validateForm()">

Анкета:<br>

Прізвище: <input type = "text" name = "surname"><br>

Ім'я: <input type = "text" name = "name"><br>

<p><b>Метод:</b><Br>

<input type="checkbox" name="meth1" value="get"> Використати get<Br>

<hr>

<input type = "submit" name = "submitBtn">

<input type = "reset" name = "resetBtn">

</form>

</body>

</html>

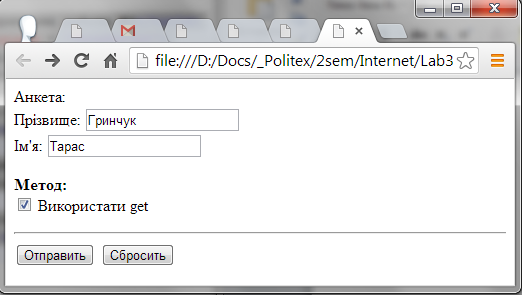


Рис. 2.3. Вигляд Web-документа завдання №3

у браузері Google Chrome

4. Побудувати веб-сторінку на вибрану тематику з полями на формі для введення паролю та імені користувача і перевірки класу користувача, вигляд у браузері – рис. 2.4. Текст:

<html>

<head>

<title>Лабораторна робота №3</title>

<script language = "JavaScript">

function validateForm() {

var x=document.forms["form1"]["usr"].value;

if (x==null || x=="") {

alert("Потрібно заповнити логін !");

return false;

}

var y=document.forms["form1"]["pas"].value;

if (y==null || y=="") {

alert("Потрібно заповнити пароль !");

return false;

}

if(x.toLowerCase().indexOf("admin") < 0) {

alert("Тільки адміністратори мають право відсилати запит на сервер !");

return false;

}

return true;

}

</script>

</head>

<body>

<form name = "form1" method="post" action="demo\_form.asp" onsubmit="return validateForm()">

Введіть логін та пароль:<br>

Користувач: <input type = "text" name = "usr"><br>

Пароль: <input type = "password" name = "pas"><br>

<hr>

<input type = "submit" name = "submitBtn">

<input type = "reset" name = "resetBtn">

</form>

</body>

</html>

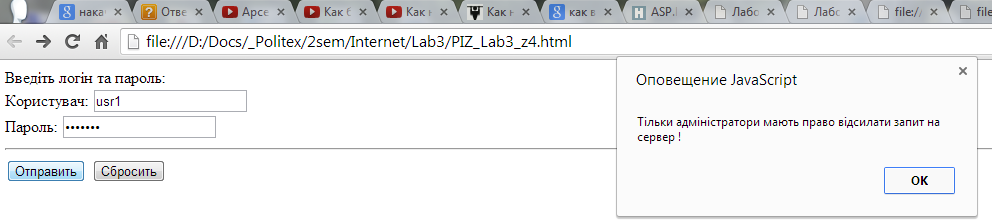
****

Рис. 2.4. Вигляд html-документа

для завдання №4 у браузері

Google Chrome

## ВИСНОВКИ

На даній лабораторній роботі я навчився створювати та HTML-форми зворотнього зв'язку, зокрема за допомогою скриптів на мові JavaScript, навчився робити різноманітні перевірки інформації на формі на коректність перед відправкою її на сервер.