**OVERVIEW & PURPOSE**

The purpose of this lab is to understand what role JavaScript plays in client and server communication. To achieve these goals, you will work with a rudimentary (not intended for use outside of the classroom) HTTP server via Node.js. The server side application will allow a user to enter a comment in the browser that will be sent to the server. The server will then save the URI and send a new dynamically created web page with an updated URI list.

Lab 7

*(Please use this*[*chapter http://eloquentjavascript.net/17\_http.html*](https://www.google.com/url?q=http://eloquentjavascript.net/17_http.html&sa=D&ust=1490815932283000&usg=AFQjCNFgw_e_jCxF-Qw048K30Xeu9bCDuQ)*of Eloquent JavaScript to complete questions 1-3.)*

1. What is the message that the client sends to the server called? What is the message that the server sends to a client called?

**The message that client sends to the server is called REQUEST. There several type of requests: GET, PUT, DELETE, POST. The message that server sends back to the client is called RESPONSE.**

1. What does the three digit code 200 mean? What do codes starting with 4 mean?

**The three digit code in the first line of the response code is the status of the response. If the code is 200, it means request succeeded. Actually any number starting with 2 means request succeeded. Usually there will be human readable string after that code, the code 200 corresponds to string “OK” in human readable format. All codes starting with 4 means request failed and there is no resource requested. When we visit nonexistent page of the web site, we usually get 404 error, which means there is no such page that server can locate.**

1. What is an HTML form? And what happens when you submit one?

**HTML form is the combinations of fields that user can enter text, choose/mark/select option and submit it to the server. When user submits the form, depending on the method of the form, it will formulate request and sends ii to the server. If method is GET, request is formulated in the URL. GET requests are used for harmless actions like search. But if your information is sensitive, then it is better to use POST requests. In POST requests information that user enters are embedded to the content of the request rather than URL. Actions that require some changes in database should use POST requests. For example when you create a new Facebook account, it uses POST request. The resource of the request is shown in action property of the form.**

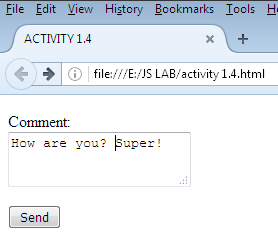
4. Please create an html file in notepad and copy and paste the html form code below. Open the file in your browser.

<form method="GET" action="example/message.html">

    <p>Comment:<br><textarea name="comment"></textarea></p>  
   <p><button type="submit">Send</button></p>                                                                                        
</form>

Type a message and click send. Copy and paste your URL. How is the exclamation point represented in the URL? Provide Screenshot

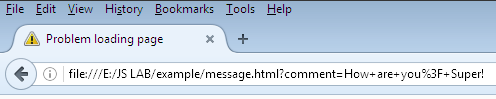
**Here is the form screenshot**

****

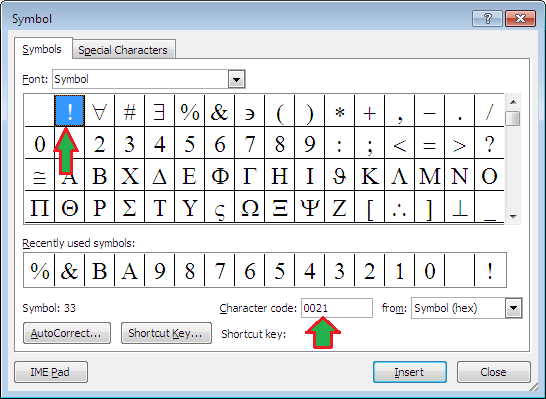
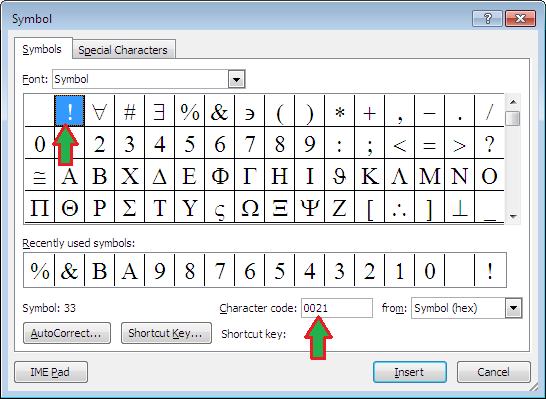
**Here is the link that was created by GET method of the form**

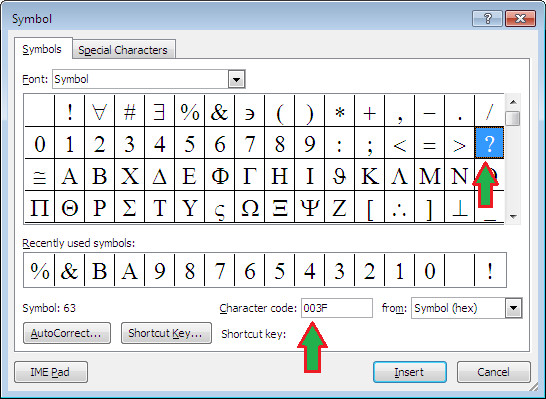
**file:///E:/JS%20LAB/example/message.html?comment=How+are+you%3F+Super%21**

**And, here is the URL screenshot.**



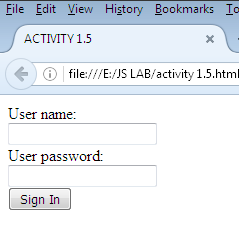
**As you have noticed, exclamation is shown as it is in the screenshot, but when I copied it and pasted to the word document, exclamation mark is coded as %21. And question mark is coded as %3F. Space is coded as %20. It turns out these are hexadecimal codes of those symbols. In MS WORD, insert symbol I have checked the codes, here are the screenshots of MS WORD’s insert symbol dialog box**

****



5. Go to this site to learn more about forms: [https://www.w3schools.com/html/html\_forms.asp](https://www.google.com/url?q=https://www.w3schools.com/html/html_forms.asp&sa=D&ust=1490815932286000&usg=AFQjCNEyA3LenXDwLr-u8qVvLQ99u3jMrA) Craft an HTML form for user login. You will need a user login box and password box. Provide your html code. Provide Screenshots[Hint: [https://www.w3schools.com/html/html\_form\_input\_types.asp](https://www.google.com/url?q=https://www.w3schools.com/html/html_form_input_types.asp&sa=D&ust=1490815932287000&usg=AFQjCNFypz750zy-QaPm1cnY_JklFoX47A)]

**This is screenshot**



**This is the HTML code**

**<!doctype html>**

**<html lang="en">**

**<head>**

**<meta charset="utf-8">**

**<title>ACTIVITY 1.5</title>**

**<meta name="description" content="JavaScript Course">**

**</head>**

**<body>**

**<form method="GET" action="activity 1.5.html">**

**User name:<br>**

**<input type="text" name="myUserName"><br>**

**User password:<br>**

**<input type="password" name="myPassword"><br>**

**<input type="submit" value="Sign In">**

**</form>**

**</body>**

**</html>**

6. Running the code below using Node.js will allow you to process form data from clients/users to render a dynamic webpage. You will use JavaScript as the server side scripting language.

Refer to the following article: [http://www.spiderwriting.co.uk/static-dynamic.php](https://www.google.com/url?q=http://www.spiderwriting.co.uk/static-dynamic.php&sa=D&ust=1490815932288000&usg=AFQjCNG_zN9IfdAnGY7OCGXx0vNv9LqgXQ) to answer the questions:

* Define a static website. Define a dynamic website?

**Static web site is written in HTML and it is what the client sees in the browser. Dynamic web site is created using languages like PHP, ASP, JSP and the content is created programmatically and usually content data is retrieved from databases.**

* What are the advantages of using a static website?

**Advantage of static web sites is that you can apply different layout and theme to each page. Moreover static web sites are loaded quicker than dynamic web sites in browsers.**

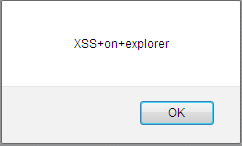
* Was your lab1 site static or dynamic?

**It was static web site.**

* What are some other popular server side languages?

**I did some research and found out there are a lot of them. The list includes the following languages C# (ASP.NET), Java (JSP), GO (from Google), Node.js, Python, PHP, Ruby, Perl.**

7. To get the server up and running using the lab computers watch the following video: [https://www.youtube.com/watch?v=ro37ql9Y718](https://www.google.com/url?q=https://www.youtube.com/watch?v%3Dro37ql9Y718&sa=D&ust=1490815932290000&usg=AFQjCNEexJWbki9-G83IRK2Zo6sRDO5kuA)  and refer to the follow notes: [https://docs.google.com/document/d/17EaRAxGIpzew9RdTpSq5LFUByGbTeL7SEcTHRiI4yqY/pub](https://www.google.com/url?q=https://docs.google.com/document/d/17EaRAxGIpzew9RdTpSq5LFUByGbTeL7SEcTHRiI4yqY/pub&sa=D&ust=1490815932291000&usg=AFQjCNE3i4aBNrCL3dY6mHxlSnLBS2xuJQ) Provide Screenshot





**<p>/index.html/index.html?comment=**

**<script>alert("XSS+on+explorer")</script>**

**/index.html</p>**

**<form method="GET" action="index.html">**

**<p>Comments:<br>**

**<textarea name="comment"></textarea></p>**

**<p><button type="submit">Send</button></p>**

**</form>**

8. Type a comment into the box and hit the send button. Open another client (browser tab) and type a command. What happens? Are the displayed URIs being saved by the client or the server?

**Entered messages keeps displaying as URL like GET method creates it.**

**I think it is being saved to the server and when another user visits this page, it retrieves the saved messages and displays it. It is dynamic web site. For saving the messages it uses comments.txt file.**

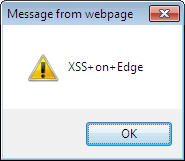
9. What is XSS? [Wiki Def](https://www.google.com/url?q=https://en.wikipedia.org/wiki/Cross-site_scripting%23Reflected_.28non-persistent.29&sa=D&ust=1490815932292000&usg=AFQjCNFdBMeSVes_2_IAzdbdTk_5lLclfw)

**XSS is acronym for Cross Site Scripting. As I understood when you open your accounts in bank or other personal information, the web site of the bank gives your browser permission to open the bank account after passing several security checks like password, code. When you open other web site, their malicious code uses permission given by your banks web site to your browser for their own malicious purposes. This is called Cross Site Scripting.**

10 . [Note: You should not try this outside of lab/only to test the security of your own websites. This is illegal to do to someone else’s website.] Open ~~explorer~~Microsoft Edge and enter the local server url: [http://localhost:8000/index.html](https://www.google.com/url?q=http://localhost:8000/index.html&sa=D&ust=1490815932293000&usg=AFQjCNGKspGGINGiSYxLldiK8xWQFwSurA) Type <script>alert("XSS on Edge")</script> and hit enter.

* What happens?

**It displays XSS+on+Edge message as shown in screenshot below**

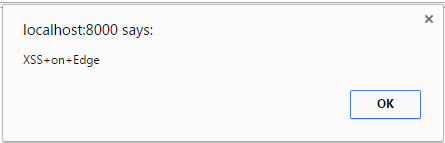


Open chrome?

* What happens?

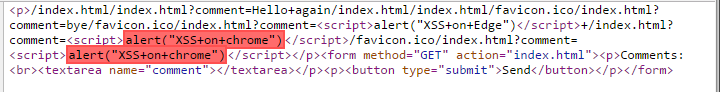
Now type <script>alert("XSS on chrome")</script>.

**Even though I typed script above, it keeps displaying my previously entered message.**



* Does XSS work? View the source code. What is highlighted?

**The message I entered is highlighted.**



 How could you write server side code to prevent such an attack?

11.  Edit the server side code so that the form you created in step 5 is displayed. Provide a screenshot of the browser displaying the form with a test username and password.

*// JavaScript source code*  
*//adapted from http://eloquentjavascript.net/20\_node.html*  
  
*//this gives your program acccess to the file system module methods for read and writing to a file \*aka file I/O*  
var fs = require("fs");  
*//this gives your program access to the Node.js's HTTP API  so that it can recieve HTTP request messages and create HTTP response messages*  
var http = require("http");  
  
*//this receives each client request method and creates and sends a response*  
var server = http.createServer(function (request, response) {  
  
   *//write the URI from the HTTP request to a file stored in the same location as this script*  
   fs.appendFileSync("comments.txt", decodeURIComponent(request.url));  
   console.log(decodeURIComponent(request.url));  
  
   *//creates the HTTP response header*  
   response.writeHead(200, { "Content-Type": "text/html" });  
  
   *//reads all of the URIs from the comment file that the new URI was just written to*  
   var str = "";  
   str = fs.readFileSync("comments.txt", 'utf8');  
  
   *//to the HTTP response message add the dynamic/updated html to include updated content*  
   response.write("<p>"+str+'</p><form method="GET" action="index.html"><p>Comments:<br><textarea name="comment"></textarea></p><p><button type="submit">Send</button></p></form>');  
  
   *//sends msg*  
   response.end();  
})  
server.listen(8000);

