Lab # 01: Introduction to Dynamic Web Content

OBJECTIVES OF THE LAB

This lab aims at the understanding of:

- *HTML*
- *HTTP and HTML*
- *The benefits of PHP, MySQL, JavaScript, CSS, and HTML*
- *The Apache Web Server*
- *About Open Source*

ABOUT HTML

HTML stands for *Hypertext Markup Language*. It uses markup tags to describe Web pages, where these tags are keywords surrounded by angle bracket like <html>. HTML tags normally come in pairs like and . The first tag in a pair is the start tag while the second Start and end tags are also known as opening and closing tags respectively.

HTML files can either be saved with .htm or .html extensions. Each HTML document describes a Web page and consists of HTML tags and plain text. Web browser (like Internet Explorer or Google Chrome) read HTML documents and display them as Web pages. The browser does not display the HTML tags instead it uses them to interpret the content of the page.

HTTP and HTML

HTTP is a communication standard governing the requests and responses that take place between the browser running on the end user's computer and the web server. The server's job is to accept a request from the client and attempt to reply to it in a meaningful way (e.g. by serving up a requested web page). A client consists of web browser and the computer on which it's running. Between the client and the server there can be several other devices, such as routers, proxies, gateways, and so on. The purpose of these devices is to ensure the requests and responses are correctly transferred between the client and server. Typically, Internet is used to send this information. A web server can usually handle multiple simultaneous connections and—when not communicating with a client—spends its time listening for an incoming connection. When one arrives, the server sends back a response to confirm its receipt.

THE REQUEST/RESPONSE PROCEDURE

At its most basic level, the request/response process consists of a web browser asking the web server to send it a web page and the server sending back the page. The browser then takes care of displaying the page as shown in Figure 1.1.

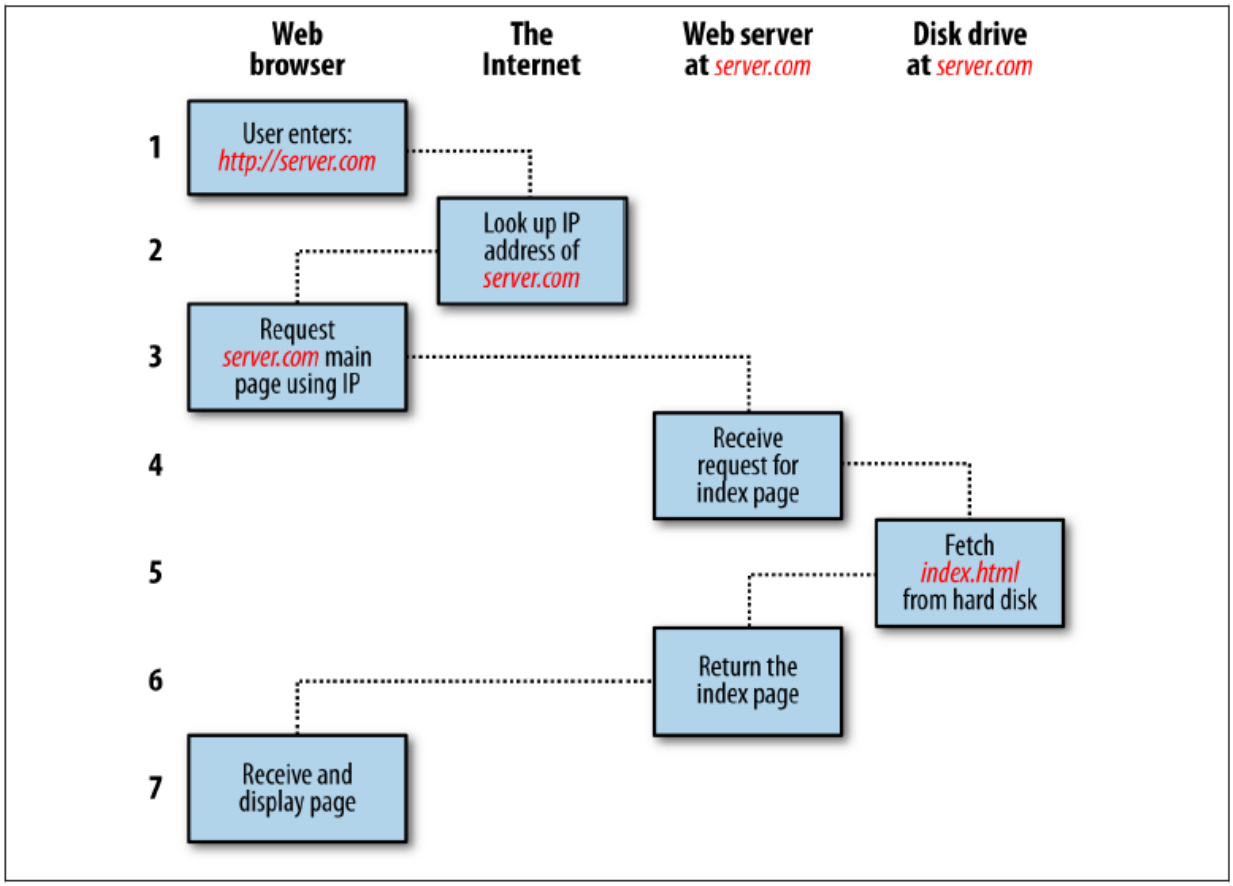


Figure 1.1 – The Basic Client/Server Request/Response Sequence

Each step in the request and response sequence is as follows:

1. You enter *http://server.com* into your browser's address bar.
2. Your browser looks up the IP address for server.com.
3. Your browser issues a request for the home page at server.com.
4. The request crosses the Internet and arrives at the server.com web server.
5. The web server, having received the request, looks for the web page on its hard disk.
6. The web page is retrieved by the server and returned to the browser.
7. Your browser displays the web page.

For dynamic web pages, the procedure is a little more involved, because it may bring both PHP and MySQL into the mix as shown in Figure 1.2.

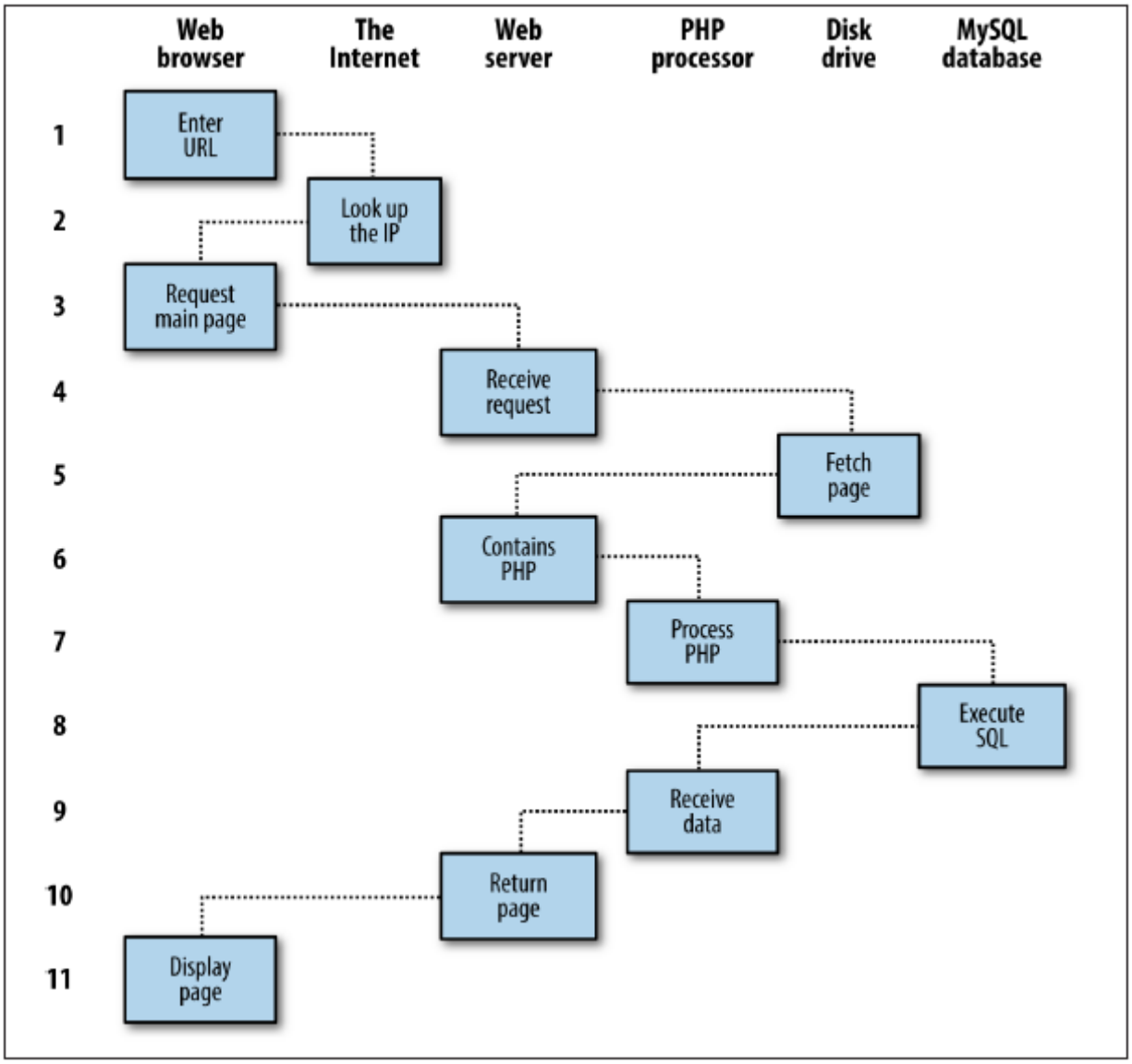


Figure 1.2 – The Dynamic Client/Server Request/Response Sequence

Here are the steps for a dynamic client/server request/response sequence:

1. You enter *http://server.com* into your browser's address bar.
2. Your browser looks up the IP address for server.com.
3. Your browser issues a request to that address for the web server's home page.
4. The request crosses the Internet and arrives at the server.com web server.
5. The web server, having received the request, fetches the home page from its hard disk.

6. With the home page now in memory, the web server notices that it is a file incorporating PHP scripting and passes the page to the PHP interpreter.
7. The PHP interpreter executes the PHP code.
8. Some of the PHP contains MySQL statements, which the PHP interpreter now passes to the MySQL database engine.
9. The MySQL database returns the results of the statements back to the PHP interpreter.
10. The PHP interpreter returns the results of the executed PHP code, along with the results from the MySQL database, to the web server.
11. The web server returns the page to the requesting client, which displays it.

THE BENEFITS OF USING PHP, MYSQL, JAVASCRIPT, CSS, AND HTML

Using PHP

With PHP, dynamic activity can be incorporated in web pages. When you give pages the .php extension, they have instant access to the scripting language. From a developer's point of view, all you have to do is write code such as the following:

```
1. <?php
2.         echo " Today is " . date("l") . ". ";
3. ?>
4. Here's the latest news.
```

The opening <?php tells the web server to allow the PHP program to interpret all the following code up to the ?> tag. Outside of this construct, everything is sent to the client as direct HTML. So the text Here's the latest news. is simply output to the browser; within the PHP tags, the built-in date function displays the current day of the week according to the server's system time. The final output of the two parts looks like this:

Today is Wednesday. Here's the latest news.

PHP is a flexible language and its construct can be placed directly next to HTML code, like this:

```
1. Today is <?php echo date("l"); ?>. Here's the latest news.
```

Using PHP, web server can be controlled by developer as per requirements. For instance, to modify HTML on the fly, process a credit card, add user details to a database, or fetch information from a third-party website, all can be done from within the same PHP files in which the HTML itself resides.

Using MySQL

MySQL is an open source, robust and exceptionally fast database management system that uses English-like commands. The highest level of MySQL structure is a database, within which you can have one or more tables that contain your data. For example, let's suppose you are working on a table called users, within which you have created columns for surname, firstname, and email, and you now wish to add another user. One command that you might use to do this is:

```
INSERT INTO users VALUES('Smith', 'John', 'jsmith@mysite.com');
```

The INSERT command here shows how to add new data in to a database. The INSERT command is an example of SQL (Structured Query Language), a language designed in the early 1970s and reminiscent of one of the oldest programming languages, COBOL. It is well suited, however, to database queries, which is why it is still in use after all this time. It's equally easy to look up data. Let's assume that you have an email address for a user and need to look up that person's name. To do this, you could issue a MySQL query such as:

```
SELECT surname, firstname FROM users WHERE email='jsmith@mysite.com';
```

MySQL will then return Smith, John and any other pairs of names that may be associated with that email address in the database.

Using PHP, all these calls can be made directly to MySQL without having to run the MySQL program or use its command-line interface. This means results can be saved in arrays for processing and can perform multiple lookups, each dependent on the results returned from earlier ones, to drill right down to the item of data required. Also, there are additional functions built right into MySQL that can be called up for common operations and extra speed.

Using JavaScript

JavaScript enables scripting access to all the elements of an HTML document. In other words, it provides a means for dynamic user interaction such as checking email address validity in input forms, displaying prompts such as "Did you really mean that?", and so on.

Combined with CSS (next section), JavaScript is the power behind dynamic web pages that change in front of your eyes rather than when a new page is returned by the server. For now, let's take a quick look at how basic JavaScript can be used:

```
<script type="text/javascript">
        document.write("Today is " + Date() );
</script>
```

This code snippet tells the web browser to interpret everything within the script tags as JavaScript, which the browser then does by writing the text **Today is** to the current document, along with the date,

by using the JavaScript function Date. The result will look something like this:

Today is Sun Jan 01 2017 01:23:45

As previously mentioned, JavaScript was originally developed to offer dynamic control over the various elements within an HTML document, and that is still its main use. But more and more, JavaScript is being used for Ajax. This is a term for the process of accessing the web server in the background. Ajax is the main process behind what is now known as Web 2.0, in which web pages have started to resemble standalone programs, because they don't have to be reloaded in their entirety. Instead, a quick Ajax call can pull in and update a single element on a web page, such as changing your photograph on a social networking site or replacing a button that you click with the answer to a question. This topic is fully covered in Lab 12.

Using CSS

With the emergence of the CSS3 standard in recent years, CSS now offers a level of dynamic interactivity previously supported only by JavaScript. For example, not only it possible to style any HTML element to change its dimensions, colors, borders, spacing, and so on, but animated transitions and transformations can be added to your web pages, using only a few lines of CSS.

Using CSS can be as simple as inserting a few rules between <style> and </style> tags in the head of a web page, like this:

```
1. <style>
2. p {
3. text-align:justify;
4. font-family:Helvetica;
5. }
6. </style>
```

These rules will change the default text alignment of the <p> tag so that paragraphs contained in it will be fully justified and will use the Helvetica font. As you'll learn in Lab 4 & Lab 5, there are many different ways you can lay out CSS rules, and you can also include them directly within tags or save a set of rules to an external file to be loaded in separately. This flexibility not only lets you style your HTML precisely, but it can also, for example, provide built-in hover functionality to animate objects as the mouse passes over them. You will also learn how to access all of an element's CSS properties from JavaScript as well as HTML.

Using HTML

As useful as all these additions to the web standards became, they were not enough for ever more ambitious developers. For example, there was still no simple way to manipulate graphics in a web browser without resorting to plug-ins such as Flash. And the same went for inserting audio and video into web pages. Plus, several annoying inconsistencies had crept into HTML during its evolution. So, to clear all this up and take the Internet beyond Web 2.0 and into its next iteration, a new standard for HTML was created to address all these shortcomings. It was called HTML5.

HTML5 is the new standard web developers now need to work to, and it will remain in place for many years to come. So learning everything you can about it now will stand you in very good stead. There's actually a great deal of new stuff in HTML (and quite a few things that have been changed or removed), but in summary, here's what it offers:

Markup

— Including new elements such as <nav> and <footer>, and deprecated elements like <font> and .

New APIs

— For example, the <canvas> element for writing and drawing on a graphics canvas, <audio> and <video> elements, offline web apps, microdata, and local storage.

Applications

— Including two new rendering technologies: MathML (Math Markup Language) for displaying mathematical formulae) and SVG (Scalable Vector Graphics) for creating graphical elements outside of the new <canvas> element.

THE APACHE WEBSERVER

In addition to PHP, MySQL, JavaScript, CSS, and HTML5, there's actually a sixth hero in the dynamic Web: the web server. Apache doesn't serve up just HTML files—it handles a wide range of files from images and Flash files to MP3 audio files, RSS (Really Simple Syndication) feeds, and so on. To do this, each element a web client encounters in an HTML page is also requested from the server, which then serves it up. But these objects don't have to be static files such as GIF images. They can all be generated by programs such as PHP scripts. That's right: PHP can even create images and other files for you, either on the fly or in advance to serve up later.

To do this, you normally have modules either precompiled into Apache or PHP or called up at runtime. One such module is the GD (Graphics Draw) library, which PHP uses to create and handle graphics.

Apache also supports a huge range of modules of its own. A list of these modules can be found on the following link: https://en.wikipedia.org/wiki/List_of_Apache_modules. Apart from PHP module, security modules are also important. Couple of other interesting module includes the Rewrite module, which enables the web server to handle a varying range of URL types and rewrite them to its own internal requirements, and the Proxy module, which you can use to serve up often-requested pages from a cache to ease the load on the server.

ABOUT OPEN SOURCE

PHP, MySQL, and Apache are all Open Source software packages. These are developed in the community by teams of programmers writing the features they themselves want and need, with the original code available for all to see and change. Bugs can be found and security breaches can be prevented before they happen.

These programs are free to use. There's no worrying about having to purchase additional licenses if you have to scale up your website and add more servers. And you don't need to check the budget before deciding whether to upgrade to the latest versions of these products.

ENDING REMARKS

The real beauty of PHP, MySQL, JavaScript, CSS, and HTML5 is the wonderful way in which they all work together to produce dynamic web content: PHP handles all the main work on the web server, MySQL manages all the data, and the combination of CSS and JavaScript looks after web page presentation. JavaScript can communicate with PHP code on the web server whenever it needs to update something (either on the server or on the web page). And with the powerful new features in HTML5, such as the canvas, audio and video, and geo-location, web pages can be made highly dynamic, interactive, and multimedia packed.

----------Task 1.1----------

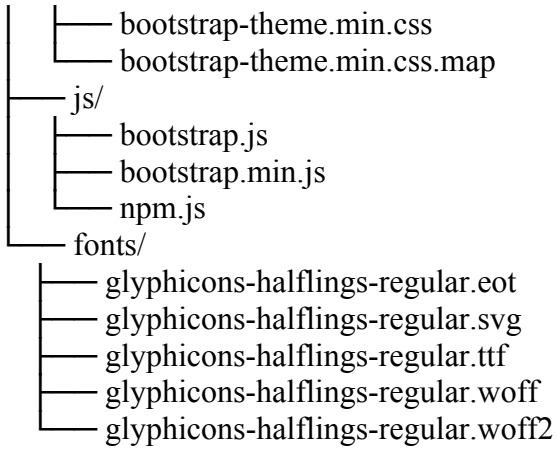
Answer the following questions:

1. What four components (at the minimum) are needed to create a fully dynamic web page?
2. What does HTML stand for?
3. Why does the name MySQL contain the letters SQL?
4. PHP and JavaScript are both programming languages that generate dynamic results for web pages. What is their main difference, and why would you use both of them?
5. What does CSS stand for?
6. List three major new elements introduced in HTML5.
7. If you encounter a bug (which is rare) in one of the open source tools, how do you think you could get it fixed?

----------Task 1.2----------

Nowadays, Twitter Bootstrap is the most popular front end framework due to its smooth, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS, and JavaScript. Its current version is 3.3.7 that is provided with this lab. The structure of pre-compiled bootstrap is as following:

```
bootstrap/
├── css/
│   ├── bootstrap.css
│   ├── bootstrap.css.map
│   ├── bootstrap.min.css
│   ├── bootstrap.min.css.map
│   ├── bootstrap-theme.css
│   ├── bootstrap-theme.css.map
```



This is the most basic form of Bootstrap i.e. the precompiled files for quick drop-in usage in nearly any web project. Examples can be found in the docs/examples/ directory. Understand the working (html & css) of following examples: **starter-template**, **signin**, **navbar**, and **cover**.

----------Task 1.3----------

What is a Framework? List top three frameworks used for PHP language. How do these help developers?

----------Task 1.4----------

What is the difference between Library and Framework?

----------Task 1.5----------

What is PHP MVC Framework? Give examples. (HINT: LARAVEL, CODEIGNITER, etc).

----------Task 1.6----------

What differences do you see in SQL and NoSQL Databases? Mention in detail. (HINT: SQL Databases include MySQL, ORACLE, SQLITE, MS ACCESS, etc; while NOSQL Databases include MongoDB, CouchDB, HBase, Cassendra, Neo4j, etc.)

----------Task 1.7----------

What is web hosting and domain? List some popular hosting websites.

----------Task 1.8----------

STARRED TASK: Make a WordPress website for topic of your choice using 000webhost.com (HINT: Follow this tutorial - https://www.000webhost.com/wordpress-tutorial)

Registration #: _____ **Name:** _____ **Date:** _____

CSE 404L – Database Management Systems Lab

LAB ASSESSMENT RUBRICS

DBMS LAB 01 – Introduction to Dynamic Web Content

Dimension	Exemplary	Acceptable	Developing	Unsatisfactory	Student Score out of 10 Marks
	10	8	6	4	
Overall Impression of Lab Report	Report is complete, well written, and organized appropriately with additional elements that enhance it. There are no spellings or grammar errors.	Report is complete, briefly written, and organized. There are few spellings and/or grammar errors.	Report is mostly complete, loosely written, and fairly organized. There are spellings and/or grammar errors.	Report is incomplete, sloppy and/or disorganized. There are many spellings and grammar errors that affect clarity.	
Submission	Report is submitted on time.	Report is submitted within 24 hours of due date.	Report is submitted within 72 hours of due date.	Report was more than 3 days overdue.	
Scientific Detail and Analysis	Descriptions of scientific terms, concepts, and theories are complete, apt, and provide detail insight in the field of databases.	Descriptions of scientific terms, concepts, and theories are mostly complete, apt, and provide insight in the field of databases.	Descriptions of scientific terms, concepts, and theories are partially complete, apt, and provide some insight in the field of databases.	Descriptions of scientific terms, concepts, and theories are either incomplete or missing, and provide minimum or no insight in the field of databases.	
Verbal Communication and Understanding	Answered clearly and accurately with sufficient knowledge.	Answered clearly and accurately with average knowledge.	Answered somewhat clearly and somewhat accurately with limited knowledge.	Answered wrongly and inaccurately with no knowledge.	

Marks: (_____+_____+_____+_____)/4 = _____

Teacher Remarks and Signature: _____