CS212 Web Programming

Final Exam Notes

History of Web Technology

- Early writing was cumbersome, hard to learn and do, and controlled largely by the ‘man’.
- Physical media was the hot area, with stone and clay winning for permanence and parchment and paper (papyrus) winning for convenience (easier to store and move around).
- The situation was pretty stable through the Middle Ages, when the Church controlled – the ‘de facto’ government.

Printing Press

- Sped up printing, but works still had to be transcribed onto press blocks.
- Prone to error – if you make a mistake, expensive to start over.
- Literature, as opposed to scholarly, religious, and governmental works are now feasible.

Movable Type

- Now we can correct our documents without starting over.
- Creation of roles – Author and/or Editor role and the Typesetter/Printer role
- Hand markup of documents begins to appear as a way for the two roles to communicate.

Rise of the Machines (no, not Skynet)

- Presses begin to get mechanized.
- Controls appear to implement typesetting rules
- These controls roughly correspond to some markup.

Digital Machines (Still Not Skynet)

- Now documents are text files
- Markup commands are embedded to handle how the text is displayed.
- General Purpose computers and digital presses begin to approach each other in capability.

Markup Language

- Commands begin to formalize with a tag format.
- Pre-processing is recognized as a type of scripting language technique.
- Documents begin to have internal links based on markup tags.
- Beginnings of complex SGML (many implementations, few cover complete specification).

Meanwhile, Networking…

- Computers begin to get networked and/or shared.
- File sharing and messaging naturally follow.
- Documents are files and also get shared.
- ARPA Net (ancestor of today's Internet) allows sharing protocols for files (ftp) and messages (mail).
- Development of document browser leads to new protocol for documents with markup.

HTML

- Subset of SGML oriented on presentation markup.
- Also allows linking to external documents via a Uniform Resource Locator (URL).
- Files are platform-independent text; it’s up to the browser to interpret them.

HTML Drawbacks

- WYSINWYG – different platforms display HTML in different ways.
- Content is static, updates must take place in the document on the server and are not reflected if you've already loaded it.
- Until extended, media types such a sound or video are not supported.
- Does not take advantage of special capabilities of a platform (fonts, etc.)

Portable Document Format

- Bundle document with free reader to create a consistent display format on any platform.
- Creation tools are not free
- WYSIWYG … at a cost

HTML Request Flow

- Browser requests a document (.htm or .html)
- Server (or local file system) responds by sending file to browser
- Browser interprets HTML tags
- Interpreted document is displayed to user

CSS

- Extends presentation capabilities of HTML
- Can be external, so one (or more) CSS can control the look and feel style of several HTML documents
- Excellent for sitewise or corporate style identity control.
- Browser processes along with HTML
- Still static.

PHP HTML Preprocessor (PHP)

- Originally Pretty Home Page Tools
- PHP is a server-side tool that takes a PHP file as input and produces HTML/CSS as output
- Because it has some conditional functionality, the output can be dynamic.
- Because it is on the server, response can be slow.

PHP Request Flow

- Browser requests a document (.php)
- Server (or local file system) gets file and converts it to HTML/CSS
- The HTML/CSS (not the PHP) is sent to the browser
- Browser interprets HTML tags/CSS
- Interpreted document is displayed to user

Multi-page PHP

- Capability to pass PHP variables in URL (Get method) or in server-stored variables (Post Method)
- Get is less secure because it is visible in the browser and TCP/IP packets.
- $_POST data can be made persistent using the SESSION functionality.

PHP Advantages

- Free!
- Dynamic!
- Fast (on the server, at least) because it does not do many of the things that higher-level languages do, such as type-checking.
- Optimized to handle simple text and numbers, perfect for web documents.
- Includes some useful libraries (for MySQL, for instance).

Speaking of MySQL

- Also free!
- Full featured but optimized for web use (if you stick with text and simple data formats)
- Support is built into PHP (making the PHP/MySQL combo very attractive)

PHP/MySQL Request Flow

- Browser requests a document (.php)
- Server (or local file system) gets file and converts it to HTML/CSS
  - MySQL requests embedded in PHP are sent to database server
  - Database server returns results
  - Results integrated into HTML/CSS output
- The HTML/CSS (not the PHP or embedded MySQL) is sent to the browser
- Browser interprets HTML tags/CSS
- Interpreted document is displayed to user

JavaScript

- Browser-side, so it is faster (no Internet lag)
- Java-like syntax, so easier for traditional programmers.
- Also moves away from complicate regular expression such as is used in PHP
- Can operate with Document Object Model and also supports browser independent applets.

JavaScript Flow

- Functions are created and linked to event handlers.
- When an event is triggered, the linked function operates and modifies the displayed page (or shows a popup)
- No direct communication with web or SQL server.
- Keeping session data requires leveraging the server (usually with PHP).

Document Object Model

- Anything between an opening and closing tag is an object. Ex. some text
- Objects (called nodes) can be arranged in a parent/child/sibling hierarchy.
- Javascript and CSS can navigate the hierarchy; PHP and CSS can leverage the elements if they are named.

eXtensible Markup Language (XML)

- You can have any tag you want
  - Tag meaning is in the document context
  - Leveraged with tool(s) used to work with the document
- Enforces a strict DOM (well-formed)
  - Single root element
  - All tags must be closed
  - No overlapping tag pairs (as opposed to embedded) text = bad!

More XML

- You can define your tags before the document (DTD).
- This DTD enforces the structure of the documents it is applied to (‘valid’ document)
- XML can be used to define data structures (database), markup tags (documents), or even configuration data or session files.
- It all depends on the tool that uses the file.

Web Design

- ‘Cool’ is in the eye of the beholder.
- Early developers were programmers, so design was often awkward and hard to use.
- Solution – add graphic designers. But that resulted in ‘pretty’ but not functional sites.
- Best results combined perspectives but also looked into deeper design issues.
- Early sites were document-driven but dynamic content changed the game.

More Web Design

- Site design should take into account the purpose of the site.
  - Sell something tangible?
  - Distribute information/support?
  - Implement a community/forum?
- Best of all took a user interface-oriented design process
- User-oriented design helps bring more users to site.
- Replaces ‘ad’ sites that inherently required other content to bring users to the site.

Yet More Web Design

- Simple user-oriented design
  - Do interface mock-ups
  - Test with a focus group of potential users
  - Adjust the interface to reflect the users
  - “Rinse and repeat”
  - Final design adds all user-desired functionality and reflects usability input from the design cycle.
- No single ‘best’ solution – best is whatever works and is usually a compromise.