EECS 485 - Web Databases & Information Systems
Winter 2010

Lectures	Mon, Wed 10:30AM - 12:00PM	**Professor**	Michael Cafarella
Location	1010 Dow	**Office**	4709 CSE
Discussion Times	Fri 10:30AM - 11:30AM	**Professor Office Hours**	Mon, Wed 12PM-1PM
Discussion Location	1010 Dow	**Professor Email**	michjc@umich.edu
GSI	Vahed Qazvinian	**GSI email**	vahed@umich.edu
GSI Office Hours	Fri 12PM-1PM		

Course Description
This capstone course is a contemporary exploration of modern web-based information systems. It will integrate concepts from multiple computer science topics used in the design, development, and deployment of web-based applications, services, and knowledge systems. While broad in scope, it will also cover several key concepts in depth, including: web networking protocols, web databases and applications, web services, performance analysis and optimization, web search, web-relevant security issues, and data exchange issues. Students will learn how to incorporate these concepts into an engineering process that includes design, analysis, development and testing, using technologies such as HTTP, XML, SQL, JavaScript, AJAX, CSS, RSS, and others.

Students will form teams to implement assignments on Linux-based Apache web servers using open-source components. These assignments will culminate in large final projects that will be demonstrated and shared at the end of the class. Each group will be able to choose its own final project, subject to approval by the instructor and GSI. At the end of this course, students will understand the science behind web-based information systems and the engineering principles for building them. This is a 4-credit course and satisfies the Software Area Kernel Requirement for MS and Ph.D. students in CSE.

Fully understanding the Web requires background from many different aspects of computer science. This course will try to bring together this disparate material and make you think about how these should work together to create a usable, efficient, and secure distributed information system. Most students will probably be familiar with a portion of the class material, but very few students will have background in all of the topics covered. (If you're among the lucky few, you might consider taking something else!)

Objectives
This course is about the design and development of information systems in a distributed environment. Its primary goal is to take a holistic view of modern web systems and their constituent technologies. By the end of this course, successful students will be able to:

- Understand how n-tiered architectures can be used to implement secure, scalable systems
- Design and develop database-driven websites and applications
- Use XML for messaging and data exchange
- Implement and incorporate web services
- Utilize AJAX to improve database-driven websites
- Use RSS to push data updates to clients

- Analyze server logs to understand system performance and user behavior
- Analyze and tune server performance
- Understand designs for modern search engines and datacenters

Prerequisites
A working knowledge of databases and SQL is required. EECS 484 is a formal requirement for this class, but if you have suitable database development experience and are willing to learn the required SQL on your own, you may take this course with the instructor's permission. EECS 485 has in the past been taken by many students who lack the standard computer science background, and we hope this tradition will continue.

Because there will be a substantial amount of programming in this class, and programming will not be a major topic of lectures, you are expected to have "programming maturity." That is, you are deeply familiar with at least one programming language that is suitable for software engineering, such as Java, C, C++, Python, etc. You are willing and able to pick up other similar languages. You can understand and use new APIs by reading the relevant documentation. Students who may be unsure about their qualifications should approach the instructor with any questions.

Requirements
Achieving the course objectives will require a significant amount of learning outside of class. Lectures will cover key topics and help integrate concepts, but will not necessarily cover all implementation details. Discussion sessions will be dedicated to development techniques, assignment details, and current lecture topics. Although significant support will be available, student teams will be required to research various technologies and development techniques to complete their assignments.

Coursework will consist of a sequence of 8 programming assignments, the last of which will be a major final project. This final project must be some kind of data-backed website, but the details and design are up to you. At the end of the class, you will deliver a short in-class presentation about your project, and leave it running for several days so people can play with your creation. You will do all programming assignments and the final project as part of a group of 3 students, though groups of 2 or 4 will be allowed in special circumstances. Your group is intended to remain fixed for the duration of the class.

An important part of the course is working within a team environment to solve hard problems. As such, students will be asked to describe and evaluate the contributions of each team member. The results will be made available to the individuals and will be factored into the final grade.

There will be occasional written review questions for lecture material that will be ungraded and for which you will be provided the answers. They are intended as a study aide.

There will be two mid-terms and a final examination.

Textbook
There is no textbook to buy; readings for this course will be provided as a set of PDF files specific to each topic and links to outside information sources. The lecture notes will also be made available shortly after each class.

However, if you want to have a supplemental text for the programming assignments, here are a few relatively inexpensive options. I think the best one is:
PHP and MySQL, 2nd Edition, by Hugh Williams and David Lane. O'Reilly, 2004.

Also good:
PHP and MySQL Web Development, 4th Edition, by Luke Welling and Laura Thomson. Addison-Wesley, 2009.

If you want more of a traditional PHP text and reference:
Programming PHP, 2nd Edition, by Rasmus Lerdorf and Kevin Tatroe and Peter MacIntyre. O'Reilly, 2006.

Some of these texts may be available for free online for Michigan students. Check out: http://proquest.safaribooksonline.com

Assignments
Programming assignments are cumulative, with subsequent assignments depending upon previous ones. Make sure not to fall behind, or you will be in serious trouble. If you have an illness, a family situation, or other emergency, figure out a fair way to manage the load with your project partners.

All projects must be performed as a team effort. All team members will get the same grade on their joint work, except under very unusual (and generally unpleasant) circumstances.

Exams
There will be a total of 3 exams. The first exam will cover topics discussed in roughly the first third of the semester, and the second exam will roughly cover the middle third. The final exam will be comprehensive. There will be no make-up exams. Make sure not to miss the mid-terms or final. The midterms will be weighted equally.

Tentative Lecture Schedule
Note that this schedule only lists the first two discussion sections, but they will generally be held each week. After the first few weeks, each discussion section will cover questions on the pending programming assignment or the lecture-based ungraded review questions.

1	Wed, Jan 6	Welcome & Web Overview	
DISC	Fri, Jan 8	Intro to Apache, MySQL	Prog1 OUT
2	Mon, Jan 11	Dynamic page generation, client-side code	
3	Wed, Jan 13	TCP/IP, network architecture	
DISC	Fri, Jan 15	Intro to PHP	Prog1 IN, Prog2 OUT
4	Mon, Jan 18	MLK Day. No Lecture.	
5	Wed, Jan 20	Sessions and Personalization I	
6	Mon, Jan 25	Sessions and Personalization II. Ruby on Rails	Prog2 IN, Prog3 OUT
7	Wed, Jan 27	Security and Cryptography I	

8	Mon, Feb 1	Security and Cryptography II	
9	Wed, Feb 3	XML I	Prog3 IN, Prog4 OUT
10	Mon, Feb 8	XML II	
11	Wed, Feb 10	**MIDTERM #1**	
12	Mon, Feb 15	Querying XML I	
13	Wed, Feb 17	Querying XML II	Prog4 IN, Prog5 OUT
14	Mon, Feb 22	Web Services, Triggers. System in depth: RSS.	
15	Wed, Feb 24	Information Retrieval I	Prog5 IN, Prog6 OUT
--	Mon Mar 1	WINTER BREAK	
--	Wed Mar 3	WINTER BREAK	
16	Mon, Mar 8	Information Retrieval II	
17	Wed, Mar 10	Web Search I. System in depth: Nutch	Prog6 IN, Prog7 OUT
18	Mon, Mar 15	Web Search II	
19	Wed, Mar 17	Logs and Data Mining	Prog7 IN, Final Project OUT
20	Mon, Mar 22	Directories and DNS. System in depth: Akamai	
21	Wed, Mar 24	**MIDTERM #2**	
22	Mon, Mar 29	Scaling and Distribution	Final Project Proposals IN
23	Wed, Mar 31	Final Project In-class "Pitches"	
24	Mon, Apr 5	Caching and Proxies I	Final Project Alpha Release IN
25	Wed, Apr 7	Caching and Proxies II	
26	Mon, Apr 12	GFS, MapReduce, Cloud Computing	Final Project Beta Release IN
27	Wed, Apr 14	System in depth: the modern datacenter	
28	Mon, Apr 19	Final Project In-class Presentations	Final Project IN

The final exam will be held during the standard scheduled exam period for this class: Wednesday, April 28, 10:30AM - 12:30PM.

Grading
Half of your grade will come from the programming assignments; more than half of this portion will come from your final project grade. The remainder will come from your performance on the midterms, the final, class participation, and peer-evaluation by your teammates.

Evaluation	**Percentage of Grade**
Programming Assignments	20
Midterm Exams	20
Final Exam	25
Class participation, peer eval	5
Final Project	30
Total	**100**

In the event there are grading errors, please bring them to the attention of the GSI or instructor promptly. Please realize that we are careful in terms of applying a uniform grading policy, and so will not be able to make changes unless you have a particular special circumstance. No regrade requests will be entertained more than two weeks past the time the graded assignment or exam was returned to you -- it is very hard for us to grade consistently if we have to go back to things we did weeks ago and no longer have the material fresh in our heads.

Due Dates
You may take a total of 4 late days total over all group programming assignments except the final project, subject to a maximum of 2 late days for any assignment. These late days are intended to account for the inevitable illnesses, family visits, demands from other classes, etc. Every additional late day beyond 4 will cost you 1% of the final grade (total for the course). Any assignment completed more than two days late will earn a zero for that assignment.

Every now and then there will be a severe emergency that cannot be covered by your late days, in which case you should approach us and ask (as early as possible). But we expect these situations to be very rare. Plan for contingencies. Allow some slack in your schedule.

Please note: *Your late days cannot be applied to the exams, or any deadline associated with the final project.*

Final Project
This class reserves a substantial portion of time for you to work on the final project. This is intended to be a major software engineering project on your part, and should result in a working system that would be genuinely useful to a general audience. Your group can choose its own final project, subject to a few guidelines. Your project:

- Must not simply reproduce a site that already exists. It must be brand-new.
- Must be non-trivial but doable within the given amount of time. The final project schedule allows plenty of time for you to think about the scope of your project and discuss it with the professor or GSI as appropriate.
- Must result in a database-backed Web application that is accessible to anyone with a general modern browser.

- Must involve a heavy back-end data element. This could mean tracking user data, or manipulating an external dataset, or most likely some combination of these components.
- Must involve an AJAX front-end that is tailored to the application.
- Should be interesting to a small user base. Facebook is a great application, but it is useless if only one person signs up. We must be able to see your project work well even if no one else joins in. (But if it is improved when other people sign up, that's fine. Indeed, it's encouraged.)
- Should keep the screen pyrotechnics to a minimum. A single sound effect might be nice. Streaming video and a soundtrack will probably not be.
- Should probably be designed to be useful, rather than entertaining. Certainly, it should not be a video game inside a web browser window. However, this line can be fuzzy: the most useful applications often have entertaining elements. Be thoughtful and somewhat restrained on this point.
- Should not include on-screen widgets from external sources (e.g., AdSense from Google includes ads on pages. There are lots of others.) If you have a compelling reason to do so, ask before you go through with it. One exception is the use of Google Maps to render geographic data - this is an extremely useful tool that is applicable to many projects. But as a rule, the work that appears on-screen should be your own.

A few ideas for a good final project:

- A class-recommendation system. The system might consume electronic course data from the university, along with user preferences and maybe information from RateMyProfessors.com, or something similar. The site should then generate tailored results for the user/student. A system that works across multiple universities, or that considers requirements of the major, would be very useful.
- A system for distributed collaborative software engineering. Google Docs allows nice collaboration between people on Word-style documents. Try to build the equivalent for programmers. This should not be a source-code control system. Think more about times when you have programmed "over someone's shoulder," and try to enable something similar with your site.
- A system for "environmental data exploration." The federal government has released a substantial amount of data on a range of topics at data.gov, including a good deal of environmental and toxicity information. Build a system that allows the user to combine real estate listings with this government data, finding houses in areas that are particularly clean or dirty. Allow users to both explore a map and do some simple data mining. (E.g., sorting listings by price on one axis, and toxicity on the other.) In general, the data available at data.gov (and similar sources for New York City, San Francisco, and elsewhere) is a great starting point for project ideas.

Keep in mind that some of the most interesting sites on the Web were created by just one or two people. This is your chance to build something great.

Field Trip!

Unlike most classes in EECS, we will have a field trip. We will be visiting a large University of Michigan datacenter located off-campus in Ann Arbor (accessible via public transportation). You will see physical infrastructure that is extremely similar, although not identical, to those used by large Web service providers like Google and Facebook. We will see the computers (thousands of them), electrical equipment, security arrangements, backup generators, and other items. It may sound a bit boring to go look at a bunch of hardware, but most people find the experience quite surprising and interesting. Attending the field trip is not mandatory, but is highly recommended. It will take place sometime in late March or early April, after the second midterm. Details to

come.

Plagiarism
You may seek advice from any source -- the web, your classmates, others. However, every line of code you write must be your own. No copying of code from other sources, whether internal or external to the class, is permitted. Third party libraries and tools may be used, with adequate attribution, provided that the use of such library or tool does not render the assignment trivial. Please consult with the GSI or professor to make sure you have approval in advance for any third party libraries or tools you may use. Violating the plagiarism rules may bring about a range of academic penalties, including (but not limited to) losing credit on the assignment, losing credit for the course, and even expulsion from the university.

Online Discussion
We will have an online discussion board for the class, reachable from the class website. If you have questions outside of class or the GSI and professor's office hours, please post them here. If you see a question you can answer, or simply have something interesting and relevant to say, please post away (keeping in mind the plagiarism rules above). We hope we can keep a lively conversation going on the board.

Accommodations for Students with Disabilities
If you think you need an accommodation for a disability, please let me know at your earliest convenience. Some aspects of this course may be modified to facilitate your participation and progress. As soon as you make me aware of your needs, we can work with the Office of Services for Students with Disabilities (SSD) to help us determine appropriate accommodations. SSD (734-763-3000; http://www.umich.edu/ sswd/) typically recommends accommodations through a Verified Individualized Services and Accommodations (VISA) form. I will treat any information you provide as private and confidential.