INST 377 - DYNAMIC WEB APPLICATIONS (Fall 2017)

Section: 0101
Meeting Days and Times: M/W/F 11:00 AM to 11:50 AM
Location: KEY 0103
Instructor: Myeong Lee
E-mail: myeong@umd.edu
Office: Hornbake South 4110A
Office Hours: Monday and Wednesday 1:00 PM to 2:00 PM

Catalog Description

Pre-requisite: INST 327 Database Design and Modeling.
Credit only granted for: INST 377 or INST 407.
This course will be an exploration of the methods, tools, and processes for developing dynamic, database-driven user interfaces and websites, which will cover an end-to-end process to build a web application. This includes acquiring, installing, and running web servers, database servers, and web applications.

Extended Course Description

This course will introduce methods and tools for developing application layers that include both front-end and back-end of a web-based system. An application layer allows users to interact effectively with databases through user interfaces. This course will cover acquiring, installing and running database servers, web servers, modules, and web applications. This course will also cover methods, skills, and processes for developing and maintaining application layers that allow end-users to interact with underlying databases through dynamic web interfaces.

Student Learning Outcomes

Upon completion of the course, students will be able to:

- Articulate the basic approaches and key development elements for building databases and dynamic user interfaces.
- Acquire, install and maintain a web server as a stand-alone component or as part of a bundled software distribution.
- Acquire, install and maintain applications that facilitate interactions between different layers of the application or site architecture.
- Build basic web interfaces that allow a wide range of users to interact with underlying databases.
- Program basic interface components that will add dynamic functionality to websites.
- Articulate the relationships between JavaScript, PHP, databases, CSS, and HTML.
- Identify security issues in dynamic web applications and develop approaches to address them.
- Maintain code versions and update servers using Git.

Course Materials

- Textbook
  - Nixon, R. (2012). *Learning PHP, MySQL, JavaScript, and CSS: A step-by-step guide to creating dynamic websites*. O'Reilly Media, Inc.
- Resources
  - W3School (https://www.w3schools.com)
  - Github (http://github.com)
  - Google Cloud (https://cloud.google.com)
  - Course material repo (https://github.com/myeong/INST377)
  - D3 (https://d3js.org/)

Course Activities

- *Homework Assignments*
  There will be five assignments over the semester, each of which will include 2 to 4 questions (except Assignment 0, which is for warming up).
  - *Scores:* The full score of each assignment will be 100 points except Assignment 0 (50 points), hence 450 points in total. This total score will be prorated to the final score at the end of the semester.
  - *Types of Assignments:* Most or all of the questions will be script coding or server configuration tasks. You will have somewhere between 1 to 2 weeks to work on and complete each assignment. The assignments are *individual work*. This means that although you may consult with your classmates and the instructor to develop general approaches to solving the coding challenges, you are expected to work individually while you build, type, test and debug the code.
  - *Notifications:* Assignment questions will be made available on Canvas.
  - *Submissions:* Completed assignments will be submitted via Github.
  - *Late Submission Policy:* Timely submission of the completed assignments is essential. Each one-day late submission subtracts 20 points from the original points, so submitting an assignment after 5 days from the deadline will result in 0 regardless of the quality. If an assignment due date is a religious holiday for you or you have any serious issues that prevent you from meeting the due date, please let the instructor know as soon as the assignment is announced, so an alternate due date can be set (this notification should be made within 24 hours after the assignment is announced. Otherwise, there will be no adjustment in the due date).
  - *Help Scores:* If an assignment is marked as “help scores available”, it means you can help or being helped by any classmates to finish the assignment. In this case, students who helped another student get extra credits when both students agree that there were help activities between the two students. Helping one student will add 5 points per help activity to his or her original points (i.e., 5 * the number of students you helped). This means, ideally, a student can earn more than the full points for a help-score-available assignment. Note that helping a classmate DOES NOT mean that a helper student does an assignment on behalf of another. It means literally helping the classmate *to understand* the process/code *fully*. If a student who was helped by another does not have *basic* understanding of an assignment, the help score will not be applied.

        - If “help scores available” is not specified, students should work on the assignment by himself/herself. Any help more than a general, high-level discussion could be regarded as plagiarism or cheating.

- *Group Project*
  Students will work in 3- to 5-student teams, and build a non-trivial web-enabled application over the semester. The final project will involve identifying an end-user need for interacting with a database, determining the requirements for the application that will facilitates the interaction, developing a deadline-oriented plan for building the application, and coding and documenting the application. Groups will be asked to find and articulate their own project topics, but they may seek the instructor’s input whilst identifying possible topics and choosing the topic to be used. Students can show preferences on group mates, but the instructor may finalize project groups based on diversity in skillsets and career goals. The instructor will provide more details on the project in the class. The followings are brief plans for conducting group projects.
    - *Mid-term Presentation:* A mid-term presentation will be administered (1) to test students’ understanding of course materials and (2) to check the progress on the group project. The mid-term presentation may include explaining particular concepts/technologies that will be used in the final deliverable, interface/DB design, and the rationale of the project.
    - *Final Presentation:* Each group is expected to
        - Introduce and justify the project
        - Demo the final website that runs on a production server (Google Cloud)
        - Explain technologies and architectures used
        - Describe development processes and strategies
        - Discuss limitations and future work.
    - *Deliverables:* Along with the final presentations, each group needs to submit
        - The URL of the final system
        - Scripts and database dump files through Github
        - Technical documentations and user manuals
        - A final report for the project.
- *Final Exam*
  A final exam will be administered to test students’ understanding of web-enabled database concepts, web interface technology concepts, development processes, as well as their script coding skills. More details will be announced a few weeks before the exam.

Grading

Course grades will be assigned based on assignments, the term project, the midterm exam, and the final exam. Scores on each component will be combined to produce a single overall score for each student as follows:

Component	**Percentage**
Assignments	20%
Mid-term Presentation	10%
Final Presentation	20%
Final Reports and Code	20%
Final Exam	20%
Class Participation	10%

Final grades will be assigned using the following categories:

A+	97-100 pts.	C	73-76.9
A	93-96.9	C-	70-72.9
A-	90-92.9	D+	67-69.9
B+	87-89.9	D	63-66.9
B	83-86.9	D-	60-62.9
B-	80-82.9	F	less than 60
C+	77-79.9		

Policy on Academic Misconduct

Cases of academic misconduct will be referred to the Office of Student Conduct irrespective of scope and circumstances, as required by university rules and regulations. It is crucial to understand that the instructors do not have a choice of following other courses of actions in handling these cases. There are severe consequences of academic misconduct, some of which are permanent and reflected on the student’s transcript. For details about procedures governing such referrals and possible consequences for the student please visit http://osc.umd.edu/OSC/Default.aspx.

It is very important that you complete your own assignments, and do not share any files or other work. The best course of action to take when a student is having problems with an assignment question is to contact the instructor. The instructor will be happy to work with students while they work on the assignments.

University of Maryland Code of Academic Integrity

The University of Maryland, College Park has a nationally recognized Code of Academic Integrity, administered by the Student Honor Council. This Code sets standards for academic integrity at Maryland for all undergraduate and graduate students. As a student you are responsible for upholding these standards for this course. It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit http://shc.umd.edu/SHC/Default.aspx.

Special Needs

Students with disabilities should inform the instructor of their needs at the beginning of the semester. Please also contact the Disability Support Services (301-314-7682 or http://www.counseling.umd.edu/DSS/). DSS will make arrangements with the student and the instructor to determine and implement appropriate academic accommodations. Students encountering psychological problems that hamper their course work are referred to the Counseling Center (301-314-7651 or http://www.counseling.umd.edu/) for expert help.

Course Schedule

The course meets three times a week for 50 minutes per session.

Session	Topics	Readings	Note
W1:1	Introduction and overview	---	
2	Local development environment; Web application stacks	*Nixon,* Ch.1 and Ch.2	
3	Git Processes (1)	https://github.com/myeong/INST377/tree/master/git	Homework 0
W2: 4 (9/4)			Labor Day Holiday
5	Git processes (2) Review of SQL	*Nixon,* Ch. 8 *Nixon,* Ch.9 (~ p.222)	
6	SQL basics Importing/exporting MySQL databases	*Nixon,* Ch.9 (p. 227 – 232)	
W3: 7 (9/11)	Linux server setup using Google Cloud	https://cloud.google.com/compute/docs/	
8	Introduction to JavaScript and DOM	*W3School JS Tutorial, "*JS Introduction*"* and "JS HTML DOM" *Nixon,* p.309 – 315 *Nixon,* p.324 – 327	HW0 Due
9	JavsScript Basics: Variables, Strings, Operators	*W3School JS Tutorial, "*JS Tutorial: JS Where To – JS Math" *Nixon,* p.316 – 321	Homework 1
W4: 10 (9/18)	JavsScript Basics: Conditionals and Functions	*W3School JS Tutorial, "*JS Tutorial: JS Random – JS Debugging" *Nixon,* p.322 – 323 *Nixon,* p.331 - 348	
11	JavsScript Basics: Objects and Arrays	*W3School JS Tutorial, "*JS Objects"	

		Nixon, Ch. 15	
12	JavsScript Basics: AJAX and JSON	*W3School JS Tutorial,* “JS AJAX” and “JS JSON” *Nixon,* Ch. 17	
W5: 13 (9/25)	PHP introduction	*Nixon,* p.35 – 39	
14	PHP Basics: Variables, Operators, Conditionals and Loops	*Nixon,* p.40 – 93	
15	Strategy and Requirements Gathering in Planning a Web Application (by Audrey Ariss, Co-founder and Researcher at The Center for Open Data Enterprise)		HW1 Due Guest Lecture Confirmed
W6: 16 (10/2)	PHP: Arrays and User-defined functions	*Nixon,* Ch. 5 and Ch. 6	
17	HTML forms	*Nixon,* Ch. 11	
18	HTML forms in PHP (the POST method)	*Nixon,* Ch. 11	
W7: 19 (10/9)	PHP-based DB Operations (Select)	*Nixon,* Ch. 11	Homework 2
20	Queries for adding, deleting and editing records through PHP	*Nixon,* Ch. 10	
21	Integrating JS for dynamic forms	*W3School JS Tutorial,* “JS Forms”	
W8: 22 (10/16)	jQuery Introduction	*Nixon,* Ch. 21 (~p.551)	
23	jQuery Events and Effects	*W3School JS Tutorial,* “jQuery Events”, “jQuery Effects”	
24	Getting server data using jQuery AJAX	*W3School JS Tutorial,* “jQuery AJAX” *Nixon,* Ch. 21 (p.551 - 555)	
W9: 25 (10/23)	Fast prototyping of dynamic interfaces using jQuery UI	https://jqueryui.com/	HW2 Due

26	Data Visualization using D3 (1): Basic charts	https://d3js.org/	
27	Mid-term Presentation (4 teams)		
W10: 28 (10/30)	Mid-term Presentation (3 teams)		
29	Five ways to make your app unusable (by Brock Fanning, Technical Architect from U.S. Department of Justice)		Homework 3 Guest Lecture Confirmed
30	Data Visualization using D3 (2): More charts	https://d3js.org/	
W11: 31 (11/6)	Styling query results using JS and CSS (Bootstrap)	http://getbootstrap.com/ *Nixon,* Ch. 20	
32	Using web APIs (1): What is API and how does it look?	Slides	
33	Using web APIs (2): Data APIs	http://api.data.gov	
W12: 34 (11/13)	Visualizing query results (1) (by Deok Gun Park, Ph.D. Candidate in Computer Science)		Guest Lecture Confirmed
35	Visualizing query results (2) (by Deok Gun Park, Ph.D. Candidate in Computer Science)		Guest Lecture Confirmed
36	Using web APIs (3): Map APIs	http://leafletjs.com/	HW3 Due
W13: 37 (11/20)	Exercises + Recap	Slides	Homework 4
38			Thanksgiving Holiday
39			Thanksgiving Holiday
W14: 40 (11/27)	Using web APIs (4): Authentications	https://developers.google.com/identity/sign-in/	
41	Behind the Scene: Introduction to data structures and algorithms	Slides	
42	Basic data structures and	Slides	

	algorithms: Sorting and Retrieving		
W15: 43 (12/4)	Model-View-Controller architecture for better code structure	Leff, A., & Rayfield, J. T. (2001). Web-application development using the model/view/controller design pattern. In EDOC 2001 (pp. 118-127). IEEE.	Homework 4 Due
44 (12/6)	Guest Lecture “Nested SQL Queries, Joins and Views in PHP” by Dr. Susanne Coates from ARHU Web and Applications		Guest Lecture Confirmed
45	Final Project Presentation (First-half of the project groups)		3 teams
46 (12/11)	Final Project Presentation (Second-half of the project groups)		4 teams
47 (12/13)	Final Exam		
12/15	Final Report Due		

This schedule is for planning purposes and may change as needed.

See ELMS/Canvas for current information and deadlines.