CS 4540: Web Software Architecture

Syllabus: Administrative Details and Course Objectives

Fall 2019

Course Staff and Schedule
Professor: H. James de St. Germain
Office: 3190 MEB
Email: germain@cs.utah.edu
Office Hours: TBD – see class page
Web Page: Canvas
TAs: EJay Guo
Zeeshan Hakim
Shaurya Sahai
TBD
Lectures: MW 4:35-5:55, Web L105
Labs: No Labs
Textbook: None

Course Overview
This course introduces the basics of “full stack” web based software development, including an introduction to front ends (i.e., browser code), back ends (i.e. server code), and databases (e.g. SQL). This course is for students who have not had previous experience in standard web technologies. It is not for students who have had internships or professional work developing full stack applications.

During the semester we will learn how to write web-based client/server applications. This will include HTML, CSS, JavaScript, and AJAX in the client browser; application programs written in C# running on the server side; and databases on the back end. Deployment on Cloud resources will also be required by all students. Additional topics will be chosen, as time allows, from: AWS, REST, roles/security/validation/authentication, OAuth, advanced JavaScript/HTML5, Selenium and web scraping/testing, accessibility, encryption, Bootstrap, CSS, and XML/JSON processing, Responsive and Reactive technologies, Google Analytics, etc.
While the course will use the C# .NET core framework, students will be required to research and write a report on a related stack or stack element, e.g., Node, Angular, React, Ruby, Java, etc. This will be a self-directed learning element of the course.

**Learning Outcomes**

Upon completion of this course students will be able to:

1. Construct web pages using modern HTML and CSS practices, including modern frameworks.
2. Define accessibility and utilize techniques to create accessible web pages.
3. Outline and utilize MVC technologies across the “full-stack” of web design (including front-end, back-end, and databases) to create interesting web applications. Deploy an application to a “Cloud” provider.
4. Describe the browser security model and HTTP; utilize techniques for data validation, secure session communication, cookies, single sign-on, and separate roles.
5. Utilize JavaScript for simple page manipulations and AJAX for more complex/complete “single-page” applications.
6. Demonstrate how responsive techniques can be used to construct applications that are usable at a variety of page sizes. Define and discuss responsive, reactive, and adaptive.
7. Construct a simple web-crawler for validation of page functionality and data scraping.

**The Way of the Web (or our learning never ends)**

It is a fact that technology (e.g., tools, techniques, languages, process, etc.) in computer science changes at a pace faster than most any other discipline. The concepts that we address in this class cover the basic foundation of delivering content via the “web”. These foundations will no doubt be valid for many years to come, but the tools and languages you use may be obsolete by the time you enter the work force (or perhaps your second job…).

**Who this course is for:**

This course is for students with minor to no experience with web technologies. Students who are looking for advanced training and techniques, who already know a basic web stack such as Java, PHP, Json, AJAX, Html, MySQL, etc., will achieve little benefit from this course. If you are looking for the most recent Ruby Rails Framework, this is not the place for you.

This course expects you to have a strong understanding of software development and versioning, problem solving skills, and self-directed work ethic. You will be expected to be a mature student who is able to solve problems on your own. This course is a large projects course and will take considerable time out of your schedule.

**Working individually and in small groups**

This is a projects course and you are responsible for learning the subject by doing and understanding the work. To this end, you will work individually or in small groups. Group projects will require X times as much effort as individual projects, where X is the number of members in the group. Students who do not contribute (or contribute less) to their group will receive lower grades. If for some reason you are not able to work in a group, speak to the professor during the first week of class. When working in a group,
you are responsible for all code/documentation/etc. created by your teammates. This means you had better review it before it is submitted (e.g., if your teammate provides code not written by themselves (e.g., taken from a web source) you will also be held accountable for this action. At the end of each project, you will be asked to discuss your group and describe how much work each partner contributed. In cases where one partner has not held up their end of the bargain, a grade penalty, up-to-and-including a zero, will be given.

**Peer Evaluations**
During this course, you will often be evaluated by your peers. When this is the case, you will be asked to show your code to other students and spend time reviewing their code as well. You will be asked to provide feedback and evaluations. These evaluations will be part of your grade.

**Course Overview**

**Prerequisites.** The minimum prerequisites for this class is CS 3500 though it is suggested that you have completed CS 3505. Further, you should be well versed in Visual Studio and C# and be able to pick up new techniques and ideas quickly. Finally, you should be able to, with limited guidance, take a problem specification and develop a software solution.

**Text.** There is no text that you have to buy. Instead, I will be assigning reading from a variety of sources on the web. It is expected that you will do the reading. At this point in your career you should be able to take responsibility for your own education and be able to learn on your own.

**Hardware/Software.** We will be utilizing “Cloud” resources. Each student will be expected to register for a “Cloud” account and maintain a working server until grades are due. Further, you will be expected to install the Visual Studio environment, the IIS web server, and a database on your own computer for testing purposes; alternatively, you may use the computer labs in Web lower. You will be given instructions on the basic setup and use of these tools, but will be expected to explore more advanced capabilities on your own.

**Lectures.** You are expected to come to all lectures, which will combine lecturing with active learning techniques. Participation across the course will be included in your grade.

**Lab Sections.** There are no labs sections for 4540.

**Consulting.** All of the course staff (instructor and teaching assistants) will be available outside of formal classes to answer your questions and help with problems. We will post the consulting schedule on the class web page as soon as it is finalized. You should also use Piazza for questions and discussions.

**Piazza.** The course will use the Piazza forum system for tracking issues and questions, as well as allowing students to communicate amongst themselves about issues. You are strongly encouraged to help your peers and to gain from their expertise. You should, of course, not post solutions to assignments; please see the section on Academic Misconduct.

**Readings.** The class web page contains a week by week calendar that will show the lecture topic and reading assignment for each lecture. Following each lecture, I will update this calendar to reflect what
was actually covered in lecture that day. By the end of the semester, the schedule will contain a record of everything we covered.

**Problem sets.** Most Wednesdays I will post a problem set on the class web page. Each problem set will consist of a collection of written and/or programming problems and will be due the following week (as noted on the assignment web page). Instructions on how to “hand in” your work will be posted on the same page.

**Participation.** On a regular basis, I will make use ask for your input to questions during class, using a mechanism such as Google Forms. Participation on a regular basis will be required.

**Exams.** There will be a midterm exam for this course. This exam will be slated for the 10th week of the course. October 30th.

**Grading.** A weighted average will be calculated based on:

- 10% course participation
- 40% your weighted homework – including projects and peer evaluation scores
- 10% final project
- 15% alternative stack paper
- 25% midterm exam

When evaluating your work, I use the University of Utah grading standards under which a B grade represents: Good performance, substantial achievement. To earn an A you will need to demonstrate: Excellent performance, superior achievement. Average achievement in the course will result in a C grade. Note: group members are usually assigned the same grade for the team’s work, but if there is a discrepancy in the group’s evaluation of your contribution, your grade may be scaled (either higher or lower) based on this feedback.

Often assignments will be evaluated on the following scale:

- Excellent (A)
- Good (B)
- Average (C)
- Needs Work (D)
- Poor/Unacceptable (little credit)

We use this scale for two purposes. First it is difficult to say what an 86 vs. an 87 means on a numeric score, but it is apparent when work is good vs. excellent. Second, the TAs will spend less time finding minor point deductions and more time giving feedback. You should expect a paragraph or two of feedback on every assignment. Should this not occur, contact the TA and then the professor.

Note: Final grades will be assigned on a curve consistent with the weighted averages of the students. That being said, you can generally expect the standard distribution, where 90 is an A-, 80 is a B-, 70 is C-, etc.
Students who wish to appeal a grade on an assignment must do so within one week of receiving the grade. You must first write a written summary of your appeal and post this on the appropriate Piazza board for this purpose; then you should directly consult with the Teaching Assistant responsible for your grading.

**Getting Help and Information**

You are responsible for all information posted on Canvas. You should use the Modules view to see a week by week summary of the course. Do not rely on the “to do” list in Canvas. Please make sure you are familiar with the course staff consulting hours, the Piazza discussion forums, the assignments, as well as examples from lecture, grades, and links pertaining to the reading and the technology that we encounter.

Do not send email to the course staff. Utilize the appropriate Piazza tag. The only exception to the no e-mail policy is if you need to communicate with me about a sensitive topic. In this case you can email me directly at: germain@cs.utah.edu or come visit me in my office.

**My Promise to You**

- I will explain why things are important and guide you on your path. (Though it is up to you to walk the path.)
- The course staff and I will answer your questions, but the answer may be a pointer to self-discovery.
- We will provide technical support, but expect you to have made a reasonable effort at "Self-Discovery".
- I will coach, and comfort, and help, and enable, and support you. When in doubt ask in class or come see me.
- I will make this one of your best semesters.

**Class Announcements**

Information and announcements for the entire class will be posted on the Canvas announcements. You are responsible for reviewing this information.

**Cooperation vs. Cheating**

Working with others on problem sets is a good way to learn the material and we encourage it. However, there are limits to the degree of cooperation that we permit.

On individual problem sets, you must limit your discussions with other students to a high-level discussion of solution strategies. Anything that you hand in, whether it is a written problem or a computer program, must be written in your own words. **Please see the Canvas link to “Misconduct/Cooperation”**.

All resources used in your project **must be cited/documentated**. For example, if you use a library you did not write, or a picture you did not take, or a code segment you did not come up with, you must cite the author/owner. When in doubt about what you are able to utilize in a project, ask the course staff.

On group problem sets, of course, you may collaborate with your partners.
When taking an exam, you must work completely independently of everyone else. Any collaboration there, of course, is cheating.

We do not distinguish between cheaters who copy other's work and cheaters who allow their work to be copied.

If you cheat, you will be given an E in the course and referred to the University Student Behavior Committee. Additionally, if you are removed from two CS courses because of academic misconduct, you will be permanently removed from the CS Major and not allowed to take any future CS courses.

If you have any questions about what constitutes cheating, please ask.

**Students With Disabilities**

Reasonable accommodation will gladly be provided to the known disabilities of students in the class. Please let the instructor know of such situations as soon as possible. If you wish to qualify for exemptions under the Americans with Disabilities Act (ADA), you should also notify the Center for Disabled Students Services, 160 Union Building.

**The TAs and You**

Teaching Assistants are there to help you overcome road blocks. Their purpose is to give you insights and pointers in a timely manner.

Consulting hours will be arranged for the TAs during which you may seek help with your specific problems. Do not feel shy about making heavy use of this service. Do not, however, expect the TA to do your work for you. The TAs are there to point you in the right direction. If you don't understand the problem, the TA will explain it, but you will still be expected to do your own work.

**What is expected of you:**

1. *You should have a well thought out question for the TA.*
2. If the problem is conceptual, you should describe your high level understanding of the problem, and then what you don't understand.
3. If the problem is technical, you should have your debugger open and pointing at the stopped program, or if not a programming problem, you should have Google open and pointing at a query describing your problem.
4. Before asking the TA for help, you should commit your current status to the revision control system. After, you have received help and found a solution, you should kmpare the new and previous versions of your code and identify the solution path.

What the TAs will do:

1. The job of the TA is to understand your issue and provide you guidance to solving the issue. The TA is not there to solve the issue, but to help you DISCOVER the PROCESS for SOLVING this and similar issues IN THE FUTURE!
2. The TAs will often point you to class readings or web resources, ask you to refer to your notes, ask you to detail what steps you have made to resolve this problem on your own, etc.
**TA Queue:**

We will be using the CADE lab CS TA Queue. You should make sure that you include a short description of your problem before adding yourself.

**In the case of failure:**

Surprise: An individual TA may sometimes not be able to track down your specific error. Should this be the case, you should utilize the Piazza forum, where the entire course staff will seek to answer the question.

Further, you should adapt your Work Plan to either a) remove the need for the technical component that will not work, b) complete a "dummy/template" implementation of the component that won't work, and then c) move on to another part of the project.

**More advice:**

There are several ways to help avoid long waits for help or long sessions of “futility”. First, make sure you have read the assignment, and have written down a short summary of what is to be achieved and how you think you can go about it. This summary should be presented to the TA to help explain what you "know" and what you "do not know". Second, feel free to talk with your peers about general programming ideas, syntax, etc. Again, please refer to the section on **Cooperation** for more information on what is/is not acceptable in this manner. Finally, and perhaps most effectively, work on the assignment ahead of time and complete it before the due date! You will find that the TAs often have more availability several days prior to the due date.