Syllabus for CS304, Databases with Web Interfaces

Welcome to what I think will be an exciting class! This page describes the structure, format and requirements of the class.

TABLE OF CONTENTS

About this Course¶

If you haven't already, please read **about this course**

Educational Goals¶

The major concepts and topics are listed in the "About" page, above. The major goals of this course are that students

1. understand those concepts and topics
2. be able to build a web application using those concepts and techniques
3. be able to work in pairs and teams to accomplish a joint goal
4. to build community

The first ones are typical of CS 304 for many years, but the last is new. In this remote semester, it will be easy to feel isolated, cut off from your fellow students. One of my goals is to fight that feeling of isolation: to try to build friendships and relationships. Please help me with this.

Instructor¶

My name is Scott Anderson (he/him). Feel free to call me Scott. My office is at *home* 😁 (formerly in E116), but the best way to reach me is by email at `scott.anderson@wellesley.edu` . As of August 17, my office hours are:

- Monday Block E1 (2:30pm - 3:45pm)
- Tuesday Block B1 (10:00am - 11:15am)
- Wednesday Block D1 (1:00pm - 2:15pm)
- Thursday Block C1 (11:30am - 12:45pm)
- Friday Block E1 (2:30pm - 3:45pm)

I love to talk with students one-on-one — it's the best part of my job — so please visit me during office hours. You don't even need to have a question!

During this remote semester, and maybe from now on, I'll set up bookable appointments (15 minute slots) during my office hours. That will allow you to have some idea whether I'm available and be able to reserve some of my time. *However*, you don't need to make an appointment in order to visit. If someone else already has that slot and nothing better is available, you can come then and hope that the other person's question takes less than 15 minutes. I'll be using a waiting room, so you'll just wait in the waiting room until I'm available. Or, contact me for an alternative time. The next section has the link.

Scott's Calendar¶

Scott's Calendar

Since my calendar will also have the class sessions on it, with zoom links, you can use that to get to our class sessions.

Tutors¶

We have only one tutor this semester, but I think she'll be excellent:

- Shreya Parjan, holding drop-ins by zoom appointment on Fridays from 11-1 ET

Prerequisites¶

CS 304 has *CS 230 Data Structures* as a prerequisite. Students with equivalent programming ability may also be admitted. If you do not have this prerequisite, contact me ASAP. You should probably drop the class.

Time Slot¶

We will have synchronized class at 9:30pm-10:45pm ET. That time slot was created for synchronized remote classes, since it's a time that works for the US and Asia (where the data shows the most students come from). Unfortunately, it doesn't work for students in Europe and Africa. I am sorry for that; please let me know if the time slot will be a problem for you.

Format¶

In the COVID-19 era, I'm teaching this course in a **flipped classroom** format. That means we will spend our class time on interaction (discussions, Q&A) and on coding. The coding will be working on homework assignments and project work. It's unlikely that you will complete the assignments solely during class, but hopefully you will make good progress on it. Furthermore, I will be available to help when you get stuck. In some ways, class time becomes office hours. (I'll also hold regular office hours.)

A flipped classroom also means that I will expect you to prepare for class *before* class. That means doing the reading, watching the videos, and taking some quizzes. I have always asked students to prepare for class by doing reading before class, but the balance is much changed now. I will be avoiding lecturing as much as possible, so that we can maximize time to work on assignments.

My concern about a flipped classroom, though, is that if a student comes to class unprepared, they may not be able to get much out of class. If this happens to you, possibly through no fault of your own, you won't be able to contribute to whatever joint work you are doing with your partner on an assignment. You won't be able to ask the pertinent questions that you would ordinarily be able to do. In fact, the best thing for you to spend class time on is probably doing the reading that you weren't able to do. Hopefully, it'll be a one-time event, and you'll be able to catch up for the next class, which is probably the next *day*. So there's not a lot of time.

The format of the course is four days a week instead of the usual two days a week. That means that deadlines for assignments come fast and furious.

Schedule¶

The current **schedule** is online. My goal is to make sure everyone is comfortable with a topic before going on, so we may spend more time on these topics than I have allocated. If so, topics will be moved later, and the last topics will be omitted. I expect we will not fall more than a couple of days behind, but we'll see.

The schedule lists:

- *readings*, which should be done before that day's class. There will usually be an associated Sakai quiz, which will be mailed out beforehand. Look for that. The readings will mention if there is an associated video.
- activities for that day of class (usually shown for brevity as "do") on the schedule. No need to read that before class, but you're welcome to if you like.
- assignments (both homework assignments and project deadlines).

Please take a look at it and make sure it's clear.

Videos¶

Most of the asynchronous content of this course is presented in webpages like this one. I will also record videos where I have a demo of a procedure or result that really requires it. My current plan is to upload those videos both to the Sakai site for this course and to my YouTube channel as unlisted videos (so you can only find them with the direct link). The links will be posted to the Sakai site in the "Resources" tool. Hopefully, you'll be able to view one or the other of those. If you are not able to, please contact me ASAP.

Policies¶

Please take a couple of minutes to read my **policies**. My policies cover questions like attendance, extensions, grading standards, collaboration and the like.

Evaluation¶

Student work will consist of homework assignments, and a semester project. They are weighted as follows:

component	weight
homework	35
project	55
quizzes	10

(In this term-based offering, I have dropped the take-home exam.)

Quizzes¶

We will be using the “quiz” feature of Sakai to administer online quizzes before most, maybe all, lecture classes. These will be a few (3-5) multiple-choice questions that are based solely on the readings for that lecture. The questions are not intended to be difficult; rather, we want to make your pre-class reading more effective and educational by making you a more engaged reader. These quiz questions will be accumulated over the semester into the equivalent of one test. (For example, if we had four questions per lecture for 25 lectures, each question would be worth 1 point.)

Part of my inspiration for the idea of using quizzes is this NY Times article entitled **To Really Learn, Quit Studying and Take a Test**

Homework Assignments¶

There will be eight homework assignments. The assignments will usually all have the same weight; but I will give additional weight to those that are more time-consuming. The **schedule** lists the assignments and due dates as currently planned.

Coding Style¶

The bulk of the grade for each assignment will be for getting the code working, but coding style is still important. Assignments will link to the following page of relevant **coding criteria**.

In the past, 70% of each coding assignment was for *functionality* (does the code work?) and 30% was allocated to style (is the code readable, secure and efficient?). This meant that working code would earn a passing score, even if the coding style was awful (in practice, it rarely was). Because of the need for quick turnaround of homework in this 6-week term, I have decided to re-structure this.

Assignments will consist *solely* of the functionality component. This means that

- they will be quick to grade, sometimes automatically, and
- you'll have a good idea of your grade even before you even submit your code, since you'll know how well it works

The "style" component will all be collected into a separate "assignment" that will be graded as soon as possible, but not necessarily immediately. Moreover, it will be graded in "mastery grading" style. That means that if you get a low grade early in the course, because you have not yet "mastered" the topic, you can show "mastery" later in the course, and the later grade will *replace* the earlier one.

I will expect you to continue to practice your mastery of good coding style, so your coding style will be graded each time and each assignment.

I would love for everyone to get an A on this aspect of the course.

Grading¶

There has been a lot of discussion recently about equity in grading policies and how that may be different from equality. As an extreme example, if I based the grade on a single high-stakes exam at the end, it would treat everyone *equally*, but might not be "fair." A student who doesn't "test well" (maybe has an anxiety condition or something else) or just has a bad day, can end up with a grade that doesn't fairly represent their knowledge of the material. Particularly in a remote offering, trying to conduct an exam in equal conditions is very challenging: one student has a room to themselve with high-speed internet and another is sharing a bedroom and poor wi-fi signal with two younger siblings).

That's the main reason that I've dropped the exam in this offering of the course. For the coding assignments, I am hoping that everyone takes advantage of the testing tools I have provided and that there are no surprises (that is, you know that your code passes 8 out of 10 of the test cases, and when I run your code it passes 8 out of 10). If there is ever a surprise, talk to me about it.

In some cases, I am willing for an assignment to be fixed and re-submitted. I want to be judicious about this, though, because a policy of unlimited do-overs favors students who have lots of time, and that isn't always equitable. (I recall a student who thanked me for imposing a time-limit on the take-home exam because they felt disadvantaged compared to other CS majors who has a lot more time available.)

Project¶

The **project** consists of phases which are not equally weighted:

- **P0 Individual Ideas (10 percent)**
- **P1 Team Proposals (10 percent)**
- **P2 Design and Plans (10 percent)**
- **P3 Draft Version (20 percent)**
- **P4 Alpha Version (20 percent)**
- **P5 Beta Version (20 percent)**
- **P6 Final Document (10 percent)**

The last two phases of the project are due at the end of the final exam period.

Note that, with homework assignments, I use a numeric grading rubric because I know exactly what I'm looking for. However, for the project phases, I'm looking for quality of work, and I prefer a letter grade scale: A, A-, B+, etc. These are recorded in Sakai as 95, 92, etc.

Pair Programming¶

There is a lot of evidence that students learn more and produce better work when working in pairs. For that reason, I am encouraging pair work for all homework assignments.

I am going to assign partners and I will assign a different partner for each homework assignment. If, for whatever reason, you are unable to connect and work with your partner, you can and should work solo. Don't lose time trying to connect with your partner. Do let me know if you've decided to work solo. In some cases, I will be able to connect two people who have been left solo by their partner, so do let me know ASAP.

If you are the person who is unable to work with your partner, please let me know. There may be good reasons that you are unable to do so; I will not punish someone who is unable to work with their partner. Just keep me informed.

Students must work in groups of three on the project. Here are some **project ideas**, but I encourage you and your partners to work out your own ideas.

No Laptop Policy¶

If we were to have in-person classes, I would apply the following policy. To promote interaction during class, I ask you not to bring your laptop, but to share a college Mac with someone else in the class. A college Mac belongs to neither person, so it's easier to switch roles/logins from time to time.

Lateness¶

All assignments are due at 11pm Eastern Time on the date indicated on the **schedule**. If you're in another time zone, hopefully it won't matter much, thanks to my zero-cost lateness policy.

Each assignment will have 4 lateness coupons, each good for a free, one-day, no-excuses extension. This will allow you to turn the assignment in up to four days late if your life has been suddenly derailed in some way.

You may not use lateness coupons to extend either of the last two project deadlines: I want to see and share your presentations, and I cannot extend anything beyond the end of exams.

While this policy may be **overly** generous, the goal is to allow you some flexibility, while acting as an incentive to keep up with the course. In general, assignments build on prior assignments, so it's important to keep up if at all possible. I've found that Wellesley students treat the deadlines seriously, and they don't need any additional pressure from me.

In extenuating circumstances (e.g. sickness, personal crisis, family problems), you may request an extension without penalty. Please try to contact me before the due date.

In no case should you ever *give up* in this course. There's usually a way forward. Talk to me about it.

Time Commitment¶

I know that focussing on schoolwork when working remotely is hard. In the pre-covid era, I would drive from home to work when I wanted to do some serious, focussed work, because I know that when I'm at home, there are so many distractions, good (my family), bad (TV) and ugly (laundry). Now I have to manage to focus on work while battling those distractions without a change of venue. So, I understand that this is hard.

Nevertheless, you should expect this course to take a fair amount of time. Indeed, there is college legislation about it. Here is the **FAQ** from the Provost's office (see the third item) about time commitment this fall. In short, you should budget for 24 hours per week for this course.

How to Succeed¶

This is a programming-intensive course. While our educational goals are to understand concepts and issues related to full-stack web applications, what we will *do* is to build one. So, being able to build working programs is crucial. For both homework assignments and the project, we will work in pairs or groups, so collaboration is important as well.

A few suggestions for being successful in this course. If you have others that you are willing to share, please do so!

- Expect to spend a lot of time. Coding always takes more time than expected. Bugs that could be squashed in five minutes sometimes take an hour. I've personally spent 30 unproductive minutes debugging the wrong program. More than once. You should expect to spend quite a few hours, outside class, on reading the lecture notes, reading the assignments, coding and debugging the programs, and the like. See the previous section.
- Read the code. In the readings, there will be English paragraphs describing concepts and techniques, but there will also be snippets and samples of code. It's easy to let your eyes glaze over as you look at code; it's tempting just to skip to the next readable paragraph. Don't do that. Force yourself to read the code.
- Practice! If you are stumbling over common, basic steps (creating a file, starting your app, etc), everything becomes painful and unpleasant. You will get better with practice, but that takes time and effort. I know you are not lazy, but you're human, and we all want to minimize time and effort. Me too.
- Be patient with yourself. Expect to fail many times before succeeding. Expect to make mistakes. Computers are really good at making us feel stupid. You're not stupid.
- Be resourceful and resilient. If my explanation doesn't make sense the first time, try re-reading. If that doesn't work, look for alternative presentations. There are tons of tutorials and references on the web. The **Mozilla Developer's Network (MDN)** is terrific. Google and Stack Overflow are useful tools. But beware that there's a lot of junk on Stack Overflow and on the web in general. Be skeptical as well.
- Be willing to ask for help. I *want* to help you. That's not just my job, it's my pleasure. (As a friend of mine says, "I teach for free; they pay me to do the grading.") I *don't* think you're stupid because you have a question; I think you want to learn. Besides, you're paying a zillion dollars to go to Wellesley in part because of the low student-teacher ratio that gives you a larger slice of my time, compared to big universities. Take advantage of that.

Here's some additional advice that I ~~stole~~ borrowed from CS 230:

- Programming is hard: Do not blame yourself for your mistakes; just give yourself more time!
- Programming is a skill: The best programmer has made every mistake in the books!
- Prepare to spend at least 5 hours outside class session for each hour in class
- When working on course materials, turn off your social media.
- Attend all lectures, quit all social media while in class.
- Do the reading and complete the weekly quizzes.
- Do the reading *before* attempting to do the homework.
- Read the homework description immediately, start thinking about it ASAP, not 1-2 days before it is due!
- Start homework early, be prepared to make mistakes. That's how you learn a skill!

The above advice is valid for both "normal" in-person class and our current remote class. A colleague wrote some advice that I thought was useful for remote classes: **his 3-page PDF on github** and **local copy of 3-page PDF**.

Honor Code¶

I believe that collaboration fosters a healthy and enjoyable educational environment, and encourage you to talk with other students about the course and to form study groups.

Unless otherwise instructed, feel free to discuss problem sets with other students and exchange ideas about how to solve them. However, there is a thin line between collaboration and plagiarizing the work of others. Therefore, I *require that you must compose your own solution* to each assignment. In particular, while you may discuss problems with your classmates, *you must always write up your own solutions from scratch*. It is *unacceptable* for two students who are not partners to turn in copies (or near copies) of each other's solutions. I will interpret such a situation as a violation of the Honor Code, and will bring it before the General Judiciary. When in doubt about acceptable levels of collaboration, please ask me for clarification.

In keeping with the standards of the scientific community, you must give credit where credit is due. If you make use of an idea that was developed by (or jointly with) others, please reference them appropriately in your work. For example, if person X gets a key idea for solving a problem from person Y, person X's solution should begin with a note that says "I worked with Y on this problem" and should say "The main idea (due to Y) is ..." in the appropriate places. It is *unacceptable* for students to work together but not to acknowledge each other in their write-ups.

When working on homework problems, it is perfectly reasonable to consult public literature (books, articles, etc.) for hints, techniques, and even solutions. However, you must cite any sources that contribute to your solution. Assignments and solutions from previous terms of this course are *not* considered to be part of the "public" literature. You must refrain from looking at any solutions from previous terms of this course. It is my policy that consulting solutions from previous terms of this course constitutes a violation of the Honor Code.

Pair Programming means that the pair of you act as an individual would as described above, so one pair can discuss the problem with another pair but cannot look at the code of the other pair.

I often distribute a “model” or “example” of what I would like you to build. That model is some running code, and having it allows you to look at the scene from all angles and try different behaviors. You are not permitted to try to reverse engineer my code, instead of trying to solve the problem from scratch.

Books¶

There are no textbooks for this course. Everything is from online readings, with occasional references on the web.

Ethics and Anti-Racism¶

Databases are a tool, and as such, are morally neutral. But like all tools, they can be mis-used, even unwittingly. Two examples, both from Ruha Benjamin's book, *Race After Technology: Abolitionist Tools for the New Jim Code*:

Databases can collect minute-by-minute information on what an employee is doing, turning a job into a high-pressure sweatshop. Apple, HP and Dell reportedly required Chinese workers to complete tasks every three seconds over a 12-hour period without speaking or using the bathroom [Benjamin, page 38].

Data can be used in seemingly neutral ways that have non-obvious, but pernicious effects. A "work for welfare" policy can impose work requirements for benefits, but allow a waiver for those in high unemployment regions.

> Taken at face value, this looks like a fair exception and sems to be race-neutral in that it could benefit poorer Americans of all backgrounds. In practice, however, people living in urban centers would not qualify because of their proximity to wealthier suburbs, which pull the overall unemployment rate down for the majority of Black urban residents. [Benjamin, page 35]

There's nothing wrong with collecting a database of unemployment data with fine-grained geographical information. But it's important to think about how it is used.

We will not have much time in this course to address these ethical issues, but I urge you to educate yourself about how software and data can be misused.

Accommodations¶

It is my goal that **every** student in this course succeed, regardless of whatever challenges they may face. That includes documented disabilities and conditions, as well as other challenges. COVID-19 has certainly presented all of us with new challenges. Some of you will overcome those challenges without my help, and kudos to you if you can, but you *don't* have to. I'm here to help, so let me know. Sometimes, I can't help, but I can at least be aware of the troubles you're facing. Feel free to let me know about those, too.

If you have a disability or condition, either long-term or temporary, and need reasonable academic adjustments in this course, please contact Disability Services to get a letter outlining your accommodation needs, and submit that letter to me. You should request accommodations as early as possible in the semester, or before the semester begins, since some situations can require significant time for review and accommodation design. If you need immediate accommodations, please arrange to meet with me as soon as possible. If you are unsure but suspect you may have an undocumented need for accommodations, you are encouraged to contact Disability Services. They can provide assistance including screening and referral for assessments. Disability Services can be reached at disabilityservices@wellesley.edu, at 781-283-2434, by scheduling an appointment online at their website **www.wellesley.edu/adr**, or by visiting their offices on the 3rd floor of Clapp Library, rooms 316 and 315.

If you have special needs of any kind, please meet with me to discuss accommodations that may be helpful to you.

Non-Discrimination¶

Wellesley College considers diversity essential to educational excellence, and we are committed to being a community in which each member thrives. The College does not allow discrimination or harassment based on race, color, sex, gender identity or expression, sexual orientation, ethnic or national origin or ancestry, physical or mental disability, pregnancy or any other protected status under applicable local, state or federal law.

If you or someone you know has experienced discrimination or harassment, support is available to you:

- Confidential reporting: Students can report their experiences to **Health Services** (781.283.2810); **Stone Center Counseling Service** (781.283.2839); or **Religious and Spiritual Life** (781.283.2685). These offices are not required to report allegations of sexual misconduct to the College.
- Non-confidential reporting:
    - You can let me know. As a faculty member, I am obligated to report allegations of sex-based discrimination to the Nondiscrimination/Title IX Office.
    - You can report directly to the **Nondiscrimination/Title IX Office**. (781.283.2451) to receive supports, and to learn more about your options for a response by the College or about reporting to a different institution.
    - You can report to the **Wellesley College Police Department** (Emergency: 781.283.5555, Non-emergency: 781.283.2121) if you believe a crime has been committed, or if there is an immediate safety risk.

Responsible Employee¶

Pursuant to Wellesley College policy, all employees, including faculty, are considered responsible employees. That means that any disclosure of discrimination, harassment, or sexual misconduct to a faculty member will need to be shared with the College's Director of Non-Discrimination Initiatives / Title IX and ADA / Section 504 Coordinator, Sonia Jurado (781-283-2451; sjurado@wellesley.edu). Students who do not wish to have these issues disclosed to the College should speak with confidential resources who are the only offices at the College that do not have this same reporting obligation. On campus, confidential resources include Health Services (781-283-2810 available 24/7), the Stone Center Counseling Services (781-283-2839 available 24/7) and the Office of Religious and Spiritual Life (781-283-2685). You should assume that any person employed on campus outside of these three confidential offices has an obligation to share information with Wellesley College through the Office of Non-Discrimination Initiatives.