Data-driven Web Applications
INFO 3300; CS 3300; INFO 5100
[SUBJECT TO CHANGE UNTIL 01/21/2020]

Instructor: Prof. Jeff Rzeszotarski, jeffrz [at] cornell.edu

Time & Location: M/W/F 11:15-12:05 in Klarman KG70

Course Prerequisites: CS 2110 & INFO 2300 (or prof. permission)

Course Website: https://jeffrz.com/info3300/

Course Policy Document: https://jeffrz.com/info3300/policies.pdf

Course Announcements Q&A: TBD

Assignment Submission: CMS (https://cmsx.cs.cornell.edu)

TBD

Graduate TAs: TBD

Undergraduate TAs: See staff directory pinned on Campuswire

Office Hours: See staff directory pinned on Campuswire

Contacts: info3300staff [at] gmail.com for Prof+Grad TAs
Post to Campuswire any content questions

Course Description

The web has become an outstanding environment for telling stories with data. This course will cover technologies for representing, modeling and displaying data in the context of interactive web pages. Practical skills for building web pages will be mixed with data mining algorithms and visualization design theory. We will use the D3 Javascript library to develop both static and dynamic visualizations, learn more about programming in Javascript, and explore web scalable vector graphics (SVG). Through design critique and formal study, we will identify the techniques visualization developers employ to create the “right” visualization for a given use case. This course introduces a number of popular data mining models and algorithms which we will incorporate into web visualizations.
Grading

- **Attendance will not be factored into your final grade**
  Attending class is absolutely critical to overall success in the course. In past iterations of this course, attendance scores have aligned with homework and exam scores. However, students have agency, and they ought to be able to judge whether they need to attend a particular session of class. As a result, there will be no formal attendance grade for this course. If any student experiences extreme personal circumstances necessitating a long-term absence during which course content will be missed, they should reach out to staff with appropriate supporting materials.

  While lecture notes will be posted online, slides are sometimes spare on text content and code may be hard to interpret without firsthand notes. TAs for the course are instructed not to re-teach missed content. Students are responsible for keeping up with material. No video or audio recordings of the course will be made available (outside of specific accommodations). Ungraded attendance will be taken throughout the term using notecards. This is a way to track how many people are coming to class and check how people are keeping up with material covered.

- **Weekly homework** (45% of final grade)
  Homework will be assigned on Mondays, due at 11:59PM ET the following Monday. Work will be turned in through CMS. At the end of the term the two lowest homework grades will be dropped. While regrade requests are permitted, they are not encouraged. See the course policy document for details on late or missing work and the exact regrade procedure.

- **Two open-ended group projects** (30% of final grade, 15% each)
  All students will complete two projects. **Group members will be randomly assigned.** Each group will be assigned a TA advisor with reports due over the project period

- **Take-home final exam** (for undergraduates in 3300) (25% of final grade)
  **OR**

  **A third group project** (for graduate students in 5100) (25% of final grade).
  (a larger interactive visualization project, similar to first two group projects)

- **Extra credit**
  There will be several opportunities for extra credit this term. If 75% of the class completes the mid-semester survey, then the class will receive a 1% bonus to their grade. A 1% bonus will be offered to students who post a neat visualization example from the web to the “Example Library” tag on Campuswire **before Spring Break**. No duplicates are allowed, so doing this earlier rather than later may be beneficial. Finally, 2% can be earned via SONA credits (0.5% for each credit up to 4 in total).
In-class work
Web programming is a complicated skill, and there’s no way to learn without practice. Class will often involve a programming problem that we will work on together. Template HTML files will be made available before class via GitHub. You are allowed to bring a laptop to class, but please sit on the upper deck of the room if you choose to use it during lectures. Given there is no formal attendance grade, we encourage students who intend to pay attention only to their laptop to do so elsewhere.

Academic Integrity
We will follow university policies as outlined in the Academic Integrity Handbook. You are encouraged to discuss homework, but each student will complete assignments alone. TAs are present to assist in your learning process but are not expected to offer specific code suggestions. Learning from other individuals’ code is an important part of programming, but for group projects the code should be the work of the group members except for standard libraries such as D3, lodash, and jQuery. Any code used in projects that was not written by the group members should be placed in separate files and clearly labeled with their source URLs. If you have benefitted from online resources (such as examples or StackOverflow answers), list the URLs in comments in your own code, even if you did not directly copy anything. Recall that some workplaces ban StackOverflow outright to avoid diluting their IP. Project work that relates to your other classes or research is encouraged, but you may not recycle assignments. There must be no doubt that the work you turn in for this class was done for this class and this class alone. This includes lecture notes and code! Integrity issues will be adjudicated at the discretion of Professor Rzeszotarski and course staff and include penalties such as referral to the Academic Integrity Board.

Blatant instances of copying lecture notes into homework will be considered integrity violations. This is a zero-tolerance policy. If you are concerned about inadvertently duplicating course code in your submissions, try reading the course notes first, closing them out, and then coding yourself in a new environment or editor. If you get stuck, switch back again but avoid having both open at the same time.

Other Course Policies
For detailed information on course policies (e.g. late work, regrades, attendance, scales, and special accommodations) please refer to the course policy document linked at the top of this document and on the course web page.