

Database-Backed Web Programming
CSE 491 s1, Spring '14
Syllabus 1/7/13 (v1)

Lecture/lab: Tu/Th 3-4:20pm, 55 Union Bldg

Instructor: C. Titus Brown, ctb@msu.edu, BPS 2228(c)
TA: Leigh Sheneman, leighs@msu.edu, BPS 2228.

Office hours: Wed evenings in Egr

Objectives:

In this course, you will learn how the Web works by working on an HTTP server, a backend Web app (including database and HTML generation), and a front-end JavaScript interface. As part of this we will discuss concepts in client-server and peer-to-peer architectures and how all of this technology works “under the hood” on today’s Internet. We’ll also discuss issues and approaches to developing software with an eye to maintainability, and learn about the practical separation of concerns in Web application stacks, from browser through server. A key part of this course will be the use of git and github. This course will be programming intensive and you should expect to either know Python or be prepared to learn it fairly well on your own.

Background: Everything will be done in Python (v2.7) on Linux, and you should have general familiarity with programming, including object-oriented programming and functions. There are no specific prerequisites, although I strongly recommend taking CSE 335 first. People with some Web programming background should still find the course interesting.

Materials: There are no required books or materials, and everything I use will be freely available online.

Course scope: The course will include discussion of the following topics, as well as practical work on many of them: advanced abstractions in Python; automated testing and code coverage analysis; network programming; HTTP protocol implementation; JavaScript and JQuery; HTML templating with Jinja2; simple Django programming (Python); Amazon Web Services and cloud computing; exceptions; software installs and systems administration on Linux; REST; mashups; cookies, authentication, security, and SSL; protecting against cross-site scripting attacks; code review; Facebook/Twitter interaction; parallelism; latency, throughput, and measurements; data sharding; node.js; MapReduce and Hadoop; social considerations and spam monitoring; logging and monitoring; and HTML 5.

Course outline:

The course will contain three major topic groupings:

HTTP server development.

Application development and database integration.

Software engineering and support technology.

The first five weeks will largely be on HTTP server development and software engineering, and the last 10 weeks will be on application development and database integration.

Homework:

There will be about 10 weeks of homeworks; each homework will be assigned on Thursday and due the following Thursday at noon, as assessed by github push time. I will grade them from 0-5, and the sum across all homeworks will count for 50% of your grade. I will not grade all homeworks each week; if you don't hear back about your homework by the following Tuesday's class, **assuming your homework was handed in on time, and there are no syntax errors in the code**, you get a 5 by default. You can always ask me to take a look at your homework before or after it's due, although I may not always be able to do so in a timely manner.

If your code contains syntax errors or fails to run under python2.7 on arctic, you will get a 0.

The remaining 50% of your grade will be in projects, which will give you the option of working on a wide range of topics. Each project will be worth one or more points, and you will need to accumulate 10 points or more; each point will be given a 0-5 grade.

I will use a curve of my own devising. It may be sparkly.

There will be anonymous quizzes in class on a regular basis to assess reading comprehension. If these start going badly, I reserve the option to use them for attendance and grading.

Collaboration:

You may collaborate with whomever you want; this includes taking code from other people's projects, swiping code from the Internet, working in groups, etc. Feel free to ask for help and debugging via the mailing list or the piazza site, and freely discuss code anywhere and everywhere. However, with the exception of code reviews by others, all of the commits in your github repository *must* belong to you or to me. If you use code from elsewhere, please cite it in the comments near to the

code. Also, regardless of where you get code, you need to understand it well enough to defend your technical choices to me or Leigh.

Attendance and computer requirement:

Attendance is required, and you must bring a computer that can connect to the Internet via both a Web browser and ssh. If you cannot come for some reason, please e-mail me *before the class starts* with your reason.