

Server-Side Curriculum

A Definitely-Not-Hastily-Written Document by Andrew Benson

Prep for Students

- Install Sublime Text, Mozilla Firefox or Google Chrome, Python, and Virtualenv (TODO: write instructions for installing virtualenv)
- Configure localhost (I believe Windows requires you to)

Assumed Knowledge

- Know how to navigate the file system in a basic way on the terminal (*nix: terminal, Windows: cmd.exe (or Cygwin if you're hardcore))
- Know how to code Python on the level of 15-110

The Project

Students will create a blog with extremely limited functionality (no authentication, no images/videos/media, fox only, final destination). An example blog is **constantly** being updated at webdevblog.herokuapp.com. Students who finish early may elect to add their own features, like deleting and editing posts.

The webapp will be written in Python using the Flask framework, and will be hosted on localhost.

Goals

1. Understand the Client-Server Model

- a. Know how the Internet is a network
- b. Know the roles of clients and servers on the Internet
- c. Know the basic meaning of a socket, connection, and port
- d. Know what HTTP is and have a basic understanding of HTTP headers
- e. Understand the request-response cycle of HTTP and the concept of REST
- f. Be able to explain what happens (HTTP-wise) when an internet browser navigates to

www.cmu.edu

2. Be able to write a simple static web server in Flask

- a. Be able to create a local web project with virtualenv
- b. Understand the reason behind having “static” and “templates” directories and their purpose
- c. Be able to write a server that serves a “hello world” HTML file
- d. Be able to run a Flask server locally via localhost:<port>
- e. Be able to write a server that serves several HTML files (home, about, contact, etc)

(TODO: make these files)

3. Be able to write a simple web app

- a. Understand the difference between GET and POST
- b. Understand why it is important to use the correct HTTP method
- c. Be able to write HTML forms that make HTTP requests
- d. Be able to write a server to correctly respond to a POST HTTP request
- e. Understand the dangers of not escaping client-provided text
- f. Understand the importance of server-side verification
- g. Be able to write server-side verification scripts
- h. Be able to write a template with Jinja2 that will fill in the correct coded text
- i. Understand how to turn debug mode on and off

4. Understand where you can go from here

- a. Understand the meaning of deployment, such as what a requirements.txt file is for.
- b. Know that Hackathons exist (and see some examples of past projects)
- c. Know that persistent data is usually kept in databases
- d. Know that cookies aren't just for eating