Web Apps Programming in Python

Hardeep Johar (hj2203@columbia.edu)

January Block Week (January 17-21). Geffen 420

Course Objective

A **Web App** is application software that runs remotely on a server on the world wide web and is delivered locally to users on their browsers. The application consists, broadly speaking, of three main software layers. A *presentation layer* delivered to the browser using HTML and JavaScript (or other browser compatible scripting language). An *application layer*, usually written in a scripting language like Python, that resides on the remote server and encapsulates the logic ("smarts") of the app. And a *database layer* where application data is stored.

Since the advent of the web and web browsers, web apps have become the single most important means of b2c and b2b communication and the goal of this class is to give you a working knowledge of what it takes to assemble the three layers into a web app. We will learn the basics of JavaScript and HTML and will use the python-based web framework, Django, to build an application. About 50% of the class time will be devoted to a group project where you will, in small groups, build a web app (assisted by TAs) that you will present to the class at the end of the week. While the course is programming heavy (though we will, briefly, review the basics of Python, some prior exposure to Python is necessary) the focus is on understanding what goes into building web applications and thinking creatively about your app rather than on mere technical perfection.

Class format

We'll do a lot of programming so be ready with your laptops. Each session will be a mix of lectures (though you should be ready to follow along) and working on your web application. TAs and I will always be on hand to help you craft out your dream app.

Prerequisites

Knowledge of the basics of python (lists, dictionaries, functions, objects) is a necessary prerequisite for this class. You must have taken either B8154: Python for MBAs or passed the Python basic qualifying exam.

Software

- **python**: We'll use the latest version of Python, 3.11 from python.org. (https://www.python.org/downloads/) or 3.9 from anaconda (https://www.anaconda.com/products/individual) (NOTE: I'll be using the anaconda version)
- **PyCharm**: PyCharm is a Python and Django development environment. You need to download the professional version of PyCharm. One year student licenses are available at https://www.jetbrains.com/community/education/#students (you'll get a license to all their products but you only need to download the PyCharm professional edition).

Topics

Django: Django (https://www.djangoproject.com/) is a Python-based web-framework that is designed to make the development and maintenance of a website as painless as possible (as long as you can write Python code!)

Database servers: Database servers are applications that make database resources accessible to other programs. We will use a database server (sqlite3 and PostgresSQL) to store content data and to record data about users who visit the web app we're building. Luckily, Django hides SQL (the language of relational databases) and we don't have to learn how to use SQL

HTML/CSS: The language of web pages. HTML is a markup language. Pieces of text are 'tagged' (bold, headings, list elements, buttons, forms) and these tags are interpreted by the browser when it renders a web page. CSS is a style sheet language that integrates with HTML to create formats for a website. Mostly, you're going to have to read this up on your own at https://www.khanacademy.org/computing/computer-programming/html-css

JavaScript: A high-level language used to make web pages interactive and is often embedded inside the html on a web page. We'll learn the basics of JavaScript, enough to add basic interactivity to our web pages.

Other stuff: User authentication, web scraping, APIs, maps, and whatever fun stuff we can fit into one week

Evaluation components

- **Quick quizzes**: Quizzes will cover basics of python and any material that we've covered in class
- **Individual assignments**: A few short programming assignments, mainly for practicing material covered in class.
- **Project**: The cornerstone of the class. You will work in small groups and the expectation is that, at the end of the semester, you'll have a working prototype of a web application that incorporates all the elements that we cover in the class.
- **Project presentation**: You will, as a group, present your application in a to-be-determined format
- **Participation and attendance**: Since a hefty chunk of class time will be devoted to group project work, attendance is mandatory (you don't want to leave your group hanging). You'll be docked points for unexcused absences.