IS289: Web Development
Spring 2019

Syllabus

Overview

Description

This course introduces students to the theory and practice of web application development. Lectures cover software architecture theory and software development methodologies. Lectures are supplemented with practical instruction that illustrates how to apply these ideas with modern tools. Students gain useful experience by working as a team on a web application for a real-world stakeholder.

Time & Place

Mondays, 9am - 12pm
April 1, 2019 - June 3, 2019
GSEIS Room 111

Prerequisites

IS271 Intro to Computer Systems & Programming (or experience with Python, SQL, and HTML)

Instructor

Joshua Gomez
Lecturer | Department of Information Studies
Head of Software Development & Library Systems | Library
UCLA
joshuagomez@library.ucla.edu

Office hours: by appointment
Goals

Upon completion of this course students should be able to:

➔ Explain the trade-offs between various application architectures
➔ Explain the benefits of modern development methods
➔ Build a basic web application from scratch

For students wishing to pursue a career in information technology, this course serves as a foundation for more advanced topics in software development.

For students pursuing a more traditional path in librarianship, the knowledge gained in this course will help them become better collaborators or managers of technology projects.

Materials

Computer

All students will need a laptop computer during class with the following tools installed:

➔ Python 3 (I recommend following the Hitchhiker’s Guide to Python)
➔ Git (Github has a simple set up guide)

Accounts

All students will need an account with the following online systems:

➔ Github
➔ Heroku

Texts

Required:

None. I am striving to make this class as inexpensive as possible.

Supplemental:


*** Wait for edition covering Django 2.2 ***
Policies

Attendance
Attendance is not recorded, but it is extremely important. The class will work as a team on a software development project. The final portion of each lecture will consist of a sprint meeting in which user stories are discussed and the implementation of those stories are assigned as homework for that week. It is vital that students are present for these discussions.

Quizzes
A quiz is given every week. No makeups are allowed for the quizzes (though an early one can be arranged for an anticipated absence). The lowest quiz grade will be dropped from each student’s average.

Project & Assignments
The class will participate as a group on a real world software development project. Students will take on the role of engineers in a software development team, lead by the instructor. The class will meet with a stakeholder in week 1 and gather requirements, generate user stories, and create a system design. Students will be assigned individual or paired work in Github Issues, which should be completed by the end of each weekly “sprint.” Students will submit their work via Pull Requests in Github.

Grading
➔ 50% Weekly sprint assignments
➔ 30% Weekly quizzes
➔ 10% Solo assignments
➔ 10% Participation

Communication & Readings
General announcements will be sent to students’ email via CCLE. Discussion of project details will primarily occur during class or on Github. Course readings are linked directly from this syllabus. Any supplemental readings will be posted on CCLE. The instructor will purchase lunch or dinner for the first student who sends an email to the instructor referring to this line of the syllabus.
Schedule

Week 1 - April 1, 2019

Lecture - Methods I (Waterfall vs Agile)
Instruction - Review of Version Control (Git & Github)
- Review of Object-Oriented Programming
Readings - Wikipedia: Software Development Process, User Stories
- Jeff Knupp: OOP in Python
- Wei Wang: Visual Explanation of Git Commands
Project - Gather requirements from stakeholder
- Create user stories
Homework - Build an Object-Oriented Queue

Week 2 - April 8, 2019

Lecture - Data Models (OOP vs RDBMS; ORMs)
- Methods II (TDD & CI; Pair Programming)
Instruction - Virtual environments & package management
- Django (models, migrations, and tests)
Readings - Wikipedia: Test-Driven Development, Object-relational mapping
- TravisCI docs
- Coveralls docs
- Django docs: overview, models, and migrations
Project - Design & implement the data model
- Write unit tests
- Setup CI pipeline
Homework - Fix Failing Tests & Write Your Own
Week 3 - April 15, 2019

Lecture - Software Architecture I (Design Patterns)
Instruction - Django (views, urls, templates)
               - HTML
Readings - Wikipedia: Software Architecture, Design Patterns, Model-View-Controller
           - Mozilla Developer Network (MDN) docs: HTML
           - Django docs: views, forms
Project - Implement list and detail pages

Week 4 - April 22, 2019

Lecture - Software Architecture II (Server-side vs Client-side computing)
           - Responsive Design
Instruction - CSS
             - jQuery
             - Bootstrap
Readings - MDN web docs: CSS
           - Wikipedia: Responsive Web Design
           - Browse jQuery tutorial
           - Bootstrap documentation
Project - Style the list and detail pages

Week 5 - April 29, 2019

Lecture - Authentication (Cryptography, Role-Based Access Control)
           - Security (OWASP Top 10; SQL Injection; XSS)
Instruction - Django (forms and auth)
Readings - Wikipedia: Public Key Cryptography, RBAC
           - OWASP: OWASP Top 10
           - MDN: HTTP Authentication
           - Django docs: views, forms, auth
Project - Implement user logins and editing forms
Week 6 - May 6, 2019

**Lecture**
- Methods III (Continuous Delivery/Deployment)
- Deployment Options

**Instruction**
- Heroku
- Let's Encrypt

**Readings**
- Wikipedia: Continuous Delivery
- Heroku docs
- Let's Encrypt docs

**Project**
- Deploy to Heroku
- Implement a CD pipeline
- Set up the site certificate

Week 7 - May 13, 2019

**Lecture**
- Optimization (Performance; SEO)
- Linked Data

**Instruction**
- Schema.org
- Asset compression

**Readings**
- Moz: page speed
- MDN: Optimization
- MDN: fast HTML
- Google: SEO Starter Guide
- Schema.org docs
- Django Compressor docs

**Project**
- Optimize the site for speed and SEO

Week 8 - May 20, 2019

**Lecture**
- Software Architecture III (APIs)

**Instruction**
- Django-Rest-Framework
- Django commands

**Readings**
- Wikipedia: Representational State Transfer
- MDN: Content Negotiation
- Django Rest Framework [docs](https://www.django-rest-framework.org/)
- Django docs: [admin commands](https://docs.djangoproject.com/en/3.2/ref指挥/admin/)

**Project**
- Implement a REST-based API
- Implement a CLI batch importer

**Week 9 - May 27, 2019**

**Memorial Day Holiday - No class**

**Week 10 - June 3, 2019**

**Lecture**
- Intro to Information Retrieval (precision vs recall, tf/idf)
- Methods IV (Containerization)

**Instruction**
- Haystack
- Docker

**Readings**
- Wikipedia: [tf-idf](https://en.wikipedia.org/wiki/Tf%E2%80%93idf)
- Solr [docs](https://lucene.apache.org/solr)
- Haystack [docs](https://haystack.readthedocs.io)
- Docker [docs](https://docs.docker.com)

**Project**
- Implement search functionality
- Create Docker files

**Week 11 - June 10, 2019 (Optional)**

**Lecture**
- Software Architecture IV (Single Page Applications)

**Instruction**
- Intro to JavaScript
- Intro to Vue.js
- Intro Webpack

**Readings**
- Vue.js [docs](https://vuejs.org)
- [JavaScript The Right Way](https://github.com/tamicade/javascript-the-right-way)
- Webpack [docs](https://webpack.js.org)

**Project**
- Implement public UI as an SPA