Objective
Provide students with the necessary skills to build server-side applications and APIs using frameworks and tools common in the industry.

Concepts
The course will cover how to build server-side web applications and APIs. We will also look at the differences between traditional server-side technologies like PHP and how it differs from Node.js, an asynchronous server-side alternative using JavaScript.

Prerequisites
ITP 300 (or sufficient experience). You should be proficient with the basics of building dynamic web pages using HTML, CSS, SQL, and some server-side technology.

Lecture
3 hours / week

Course Structure
The first half of the course will cover the fundamentals of building traditional server-side rendered web applications and the PHP framework, Laravel. The second half of the course will cover building APIs using Node.js (server-side JavaScript) and how asynchronous programming differs from synchronous programming.

Required Reading
*PHP Object Oriented Solutions by David Powers: Apress, 2013 – There is a free version provided by USC. Visit [this link](#). Click the link next to Electronic Access that says "» SpringerLink - An electronic book accessible through the World Wide Web; click for information". Log in with your USC credentials, and then you can download the book in PDF format.*

You can also find used copies of the book on Amazon for very cheap.

Grading
Assignments: 25%
Labs: 10%

*Assignments and labs are due the following week at midnight.*

Class participation and attendance: 10%
Exam: 20%
Individual final project: 35%

Final course grade is determined by standard formulas:

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100% - 93%</td>
</tr>
<tr>
<td>A-</td>
<td>92% - 90%</td>
</tr>
<tr>
<td>B+</td>
<td>89% - 87%</td>
</tr>
<tr>
<td>B</td>
<td>86% - 83%</td>
</tr>
<tr>
<td>B-</td>
<td>82% - 80%</td>
</tr>
<tr>
<td>C+</td>
<td>79% - 77%</td>
</tr>
<tr>
<td>C</td>
<td>76% - 73%</td>
</tr>
<tr>
<td>C-</td>
<td>72% - 70%</td>
</tr>
<tr>
<td>D+</td>
<td>69% - 67%</td>
</tr>
<tr>
<td>D</td>
<td>66% - 63%</td>
</tr>
<tr>
<td>F</td>
<td>62% and below</td>
</tr>
</tbody>
</table>

Policies
It is the responsibility of the student to make sure assignments, labs, and the final project are turned in on time via GitHub.

Academic Integrity
The use of unauthorized material, communication with fellow students during an examination, attempting to benefit from the work of another student, and similar behavior that defeats the intent of an examination or other class work is unacceptable to the University. It is often difficult to distinguish between a culpable act and inadvertent behavior resulting from the nervous tension accompanying examinations. When the professor determines that a violation has occurred, appropriate action, as determined by the instructor, will be taken.

Although working together is encouraged, all work claimed as yours must in fact be your own effort. Students who plagiarize the work of other students will receive zero points and possibly be referred to Student Judicial Affairs and Community Standards (SJACS).

All students should read, understand, and abide by the University Student Conduct Code listed in SCampus, and available at: http://www.usc.edu/student-affairs/SJACS/nonacademicreview.html

Students with Disabilities
Any student requesting academic accommodations based on a disability is required to register with Disability Services and Programs (DSP) each semester. A letter of verification for approved accommodations can be obtained from DSP. Please be sure the letter is delivered to me (or to your TA) as early in the semester as possible. DSP is located in STU 301 and is open 8:30 a.m. - 5:00 p.m., Monday through Friday. The phone number for DSP is (213) 740-0776.
Course Outline

1/9
Class Introduction
Traditional vs. API driven web applications
Installing PHP and SQLite
Intro to Git and Github
Reading: PHP Object Oriented Solutions – Chapter 1 & 2

1/16
HTTP lifecycle
Database-driven web pages review
SQL joins
PHP Data Objects (PDO), Prepared Statements, and Parameter Binding
Deployment with a Platform-as-a-Service (PaaS)
Assignment

1/23
Object Oriented Programming in PHP
Classes, Inheritance, Statics, Namespaces
Lab

1/30
Model-View-Controller (MVC)
Laravel – Routes, Controllers, Query builder, and Views
Deployment with a Platform-as-a-Service (PaaS)
Assignment

2/6
Laravel – CRUD, flash messages, data validation
Assignment

2/13
Laravel - Object Relational Mapping (ORM)
Assignment

2/20
Laravel - Middleware and Authentication
Lab

2/27
Exam

3/6
Intro to Node.js and NVM
Asynchronous programming with callback functions
Building an API with Express
Deployment with a Platform-as-a-Service (PaaS)
Assignment

3/13
Spring Break
<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>3/20</td>
<td>Asynchronous programming with promises</td>
</tr>
<tr>
<td></td>
<td>Object Relational Mapping in Node</td>
</tr>
<tr>
<td></td>
<td><strong>Assignment</strong></td>
</tr>
<tr>
<td>3/27</td>
<td>Working with APIs using OAuth 2 Client Credentials grant type</td>
</tr>
<tr>
<td></td>
<td><strong>Lab</strong></td>
</tr>
<tr>
<td>4/3</td>
<td>Asynchronous programming with async/await</td>
</tr>
<tr>
<td></td>
<td>The Koa framework</td>
</tr>
<tr>
<td></td>
<td><strong>Lab</strong></td>
</tr>
<tr>
<td>4/10</td>
<td>Logging in with a 3rd party with OAuth 2 Authorization Code</td>
</tr>
<tr>
<td>4/17</td>
<td>TBA</td>
</tr>
<tr>
<td>4/24</td>
<td>TBA</td>
</tr>
<tr>
<td></td>
<td><strong>Final Project due Sunday 4/29 at midnight</strong></td>
</tr>
<tr>
<td></td>
<td>The final project can be on any topic of interest to you. Requirements will be sent out mid semester.</td>
</tr>
</tbody>
</table>