

# **CSCI 3800: Web API Technologies**

## **Course Syllabus**

### **Spring, 2017**

**Instructor:** Shawn McCarthy  
**Email:** shawn.mccarthy@ucdenver.edu  
**Phone:** 303-349-9745  
**Office:** North 3202  
**Office Hours:** 530pm-630pm

**Catalog Data:** Special Topics in JavaScript Web technologies for front-end development and back-end development. Building a full end to end solution with a mobile or web front-end, Web API and NoSQL database

**Co-requisites:** N/A

**Prerequisites:** CSCI 2421-3 Data Structures & Program Design

**Note:** *Each student must sign and return the attached Prerequisites Agreement form to receive any credit for any assignment or exam. If this form is not returned by the 1<sup>st</sup> week, the student will be administratively dropped from the course.*

#### **Expected Knowledge at the Start of the Course:**

- Data structures for simplifying algorithm design

#### **Expected Knowledge Gained at the end of the Course:**

- Solid understanding of JavaScript, NodeJS, Development of Web APIs, NoSQL Databases and AngularJS/Ionic in a complete solution

#### **ABET Assessment Criteria:**

- **(k)** An ability to apply design and development principles in the construction of software systems of varying complexity

#### **Course Objectives:**

**Textbook:** <https://leanpub.com/mean-machine>

**Topics:** Web APIs (Design, Best Practices and Development). JavaScript, NodeJS, TypeScript, Angular2, Ionic (Mobile)

**Course Outline:**

Lecture/ Week	Date	Topic	Reading	Assignments
1	Jan 18th	Syllabus SOAP/WSDL XML and XML Schemas	Introduction to Primers	
2	Jan 24 <sup>th</sup>	HTTP AJAX (Asynchronous JavaScript and XML)	MEAN Thinking to Getting Started and Installation	
3	Jan 31st	JavaScript/jQuery/JSON Pick Teams	Starting Node	<b>Hw1 Due</b>
4	Feb 7 <sup>th</sup>	NodeJS	Routing Node Applications to Build a RESTful Node API	
5	Feb 14 <sup>th</sup>	Authorization Basic Auth, OAuth	Node Authentication	
6	Feb 21 <sup>st</sup>	API Design / Swagger / Best Practices		<b>Hw2 Due</b>
7	Feb 28 <sup>th</sup>	API Management		
8	March 7 <sup>th</sup>	BaaS / MongoDB	Using MongoDB	<b>Hw3 Due</b>
9	March 14 <sup>th</sup>	Policies (limits, throttling)		
10	March 21 <sup>st</sup>	<b>Spring Break</b>		
11	March 28 <sup>th</sup>	Analytics		<b>Hw4 Due</b>
12	April 4 <sup>th</sup>	TypeScript/Angular2	Starting Angular to Animating Angular Applications	
13	April 11th	UX <b>Hw5 Due</b>	MEAN Stack Application Structure to Angular Authentication	<b>Hw5 Due</b>
14	April 18 <sup>th</sup>	Mobile Ionic		
15	April 25 <sup>th</sup>	IoT, Project Days		
16	May 2 <sup>nd</sup>	Project Presentations		

**Grading Policy:**

Homework 65%

Projects 35%

**Notes:** UCD Code of Honor as in the catalog:
[http://catalog.ucdenver.edu/content.php?catoid=6&navoid=530&returnto=search#Academic\\_Honor\\_Code\\_and\\_Discipline\\_Policies](http://catalog.ucdenver.edu/content.php?catoid=6&navoid=530&returnto=search#Academic_Honor_Code_and_Discipline_Policies)



## **Projects**

### **Scenario 1-** Deny payment transaction based upon IP address lookup

- Create a proxy with a payment API such as eCommerce. Need to require IP address of the payment accepting client to be included in the transaction.
- Create an inline policy that takes IP address, performs a location lookup and based upon approved location allows the transaction to be processed.
- If denied decline the transaction with a meaningful error message.
- Also store the location where the transaction originated from for future use.

### **Scenario 2** - Transaction roundup to charity

- Create a payment enabled mobile application (maybe Applepay) using eComm Restful API.
- Before capturing the transaction ask if the user wants to round up to the closest dollar and donate the change to a charity of choice.
- Create summary page for mobile app to show how much the user has donated to charity.
- Also need a method where charities can be made available to donate to.

### **Scenario 3** - Commerce application intelligence

- Implement a proxy designed to inspect the transaction payload and store the information without latency impact.
- Needs to be configurable and cannot store any sensitive transaction information such as card number, expiration or CVV.
- Implement proxy to access the stored information if stored in app services.