

CIS 311: Interactive Web Development

California State Polytechnic University, Pomona

Summer 2017

Instructor: Stefan Bund, MS
Office: Building 164 Room 3017
E-mail: slbund at cpp.edu
Phone: 949.433.7487
Office Hours: Tuesday, Thursday 4 - 5:50 pm, office above

1. Class Meetings and Office Hours, Holidays

Summer Term Dates

June 20th until August 24. Final Exam week is August 28-31, following the final week of classes.

Class meeting Times

Tuesday, Thursday from 6 pm until 6:50 pm

Class Location

Building 163, room 1020

Holidays for Summer 2017

Tuesday July 4th

2. Course Description

From our catalog:

The Design and development of web applications for business. Principles and applications of modern website design. Use of client-side scripting for website dynamics and interactivity. Development of server-side scripts for three-tier web applications. 4 units lectures/problem solving.

Instruction will focus upon core protocols supporting browser markup and web-based applications: html 5, CSS, javascript, server side scripting, cloud infrastructures and internet-facing data bases. The course entails a hands-on basis for learning, with deliverables in the form of projects and programming exercises. Students should use the course as a means to discover a potential major (application development) while gathering deep understanding about the security and infrastructure demands of employee and public-facing web applications. Additional discussion explores the corporate strategic impact of web development, and organizational/cultural change.

3. Learning Objectives

Students successfully completing this course should have acquired the ability to:

- Understand different types of web applications and how they work.
- Analyze and translate user needs and requirements into a software architectural model.
- Create wireframes and prototypes of user-centered and SEO-friendly web sites

- Create structure and content for web sites using standards-compliant HTML and HTML 5.
- Create styles for web sites using Cascading Style Sheets (CSS).
- Interact with users using client-side Javascript.
- Understand how ASP.NET works to enable dynamic contents and web applications.
- Control the flow of scripts, use conditional statements, loops, strings and arrays, write reusable modules in Visual Basic.
- Collect, validate and process information entered by users via web forms.
- Build dynamic web site that connect to a text file and database, insert, present and modify data in sophisticated ways.
- Understand the basics of web analytics and optimization, including the role of content, link building, social media marketing, usability, accessibility, other search engine and conversion optimization tactics.

4. Prerequisites

A minimum grade of C (2.0) in CIS 304, 305

5. Textbook and Software

Recommended Textbooks:

Responsive Design text at our bookstore

Required Web Development Software:

For client-side development: A text editor. Options include: Notepad (plain mode, default on all Windows), TextEdit (plain mode, default on all Mac) notepad ++ (freeware), and Dreamweaver (student discount available at bookstore). Eclipse is also excellent, especially with the JSDT installed. Other, newer javascript IDEs are also acceptable, according to your personal preferences.

For server-side development: Microsoft Visual Studio 2017 (free and available on CIS dept website -> MSDNAA Download) with built-in MS SQL server. Please acquire this tool by the final two weeks of the class, when we will produce a product using ASP.NET.

6. Exams, Projects and Assignments

Exams: One final exam during exam week. No midterm.

Quizzes. Each week one quiz is given on Thursday at the end of the class period. In cases where the student does not score above 50% on the quiz, the instructor will inspect a programming project, posted as a one minute youtube video on a blog owned by the student. **This programming assignment is called a microproject.** For each microproject, the following rules apply:

- if you wish to submit a microproject, your quiz must be 50% or below.
- submit an email to Stefan, at the above email. In your email, include the URL of the blog post where you have posted your video for the microproject.

- the grade on your microproject replaces the quiz grade.
- the email with the URL to your blog is due on the **Tuesday of the following week by 6 pm.**
- the late policy below will apply to microprojects.

Make-up policy: There will be no make-up exams except for serious and compelling reasons that are substantiated with formal documents. For example, medical cases have to be substantiated with valid doctor or hospital note stating that the student is too ill to attend the exam. Athletic events are acceptable also, and will require documentation from the team, per our university norms.

Late assignments or projects: Late assignments or projects submitted within one week after due date and time will be penalized 50%. Late assignments or projects submitted more than one week late will not be accepted. If the student submits an assignment or project late, the student is responsible to send an email to notify the instructor upon submission; otherwise the assignment or project will not be graded.

Tutoring: For free tutoring on campus, contact the Learning Center in the library.

7. Grading

Grade	Percentage
A	93.00-100.00
A-	90.00-92.99
B+	87.00-89.99
B	83.00-86.99
B-	80.00-82.99
C+	77.00-79.99
C	73.00-76.99
C-	70.00-72.99
D+	67.00-69.99
D	63.00-66.99
D-	60.00-62.99
F	0-59.99

For a detailed breakdown of assignments, weight and their details, please see section 11, 'Tentative Course Schedule and Graded Items.'

8. Class Communication [IMPORTANT]

E-mail: All emails must be sent to the instructor with a Cal Poly email account, must be signed with the student's first and last name, and must have "CIS 311" in the subject line, or it may not be read or responded. Please consult the syllabus before sending emails, especially in the area of microprojects. **Messages sent through Blackboard will not be read.**

Assignment submission

All assignments are to be hosted on Amazon Web Services (AWS). A youtube video is to be produced, detailing your work, lasting no more than one minute, with code displayed at all times during the video. A link to the video is to be posted to a blog (blogger, Tumblr, wordpress, other).

Create your blog during week one of the course, then submit the link to the home page of your blog to the instructor, using this link:

- <https://goo.gl/forms/Mjz59CZbJmUgdaKx2>
- SUBMIT THIS ONLY ONCE, PLEASE :)

To review

1. Host your micro project on AWS
2. record and post your microproject's video on youtube
3. post a link to your video in your blog, or embed the youtube in your blog (best)
4. repeat, weekly for each micro project, due at the schedule below

Github: Course materials including announcements, lecture presentations, assignments, projects documents, classroom exercises and solutions, and grades will be posted on Github. All graded assignments and projects will be visible in Blackboard Gradebook. Our github will be:

- <https://github.com/stefanbund/311>

Refer to the Github page above for all homework, learning materials, lesson plans and other details critical for your success in 311.

Subject to Change: This syllabus and class schedule are subject to change. If the student is absent from class, it is the student's responsibility to check on announcements made and make up the work while absent.

9. Course Policies

Classroom environment: The classroom is a special environment in which students and faculty come together to promote learning and growth. It is essential to this learning environment that respect for the rights of others seeking to learn, respect for the professionalism of the instructor, and the general goals of academic freedom are maintained. Student conduct which disrupts the learning process shall not be tolerated and may lead to disciplinary action and/or removal from class.

Using laptops, cellphones and other electronic devices:

- Using laptops during the class for anything other than this class, personal conversations, talking or texting on cell phones or other distracting behavior are prohibited.
- As a courtesy to all, please turn off all cell phones and pagers during class. If the student needs to be reached for family medical or significant work-related issues, the student must present evidence to the instructor before the class starts.
- Absolutely no cellphones or other electronic devices may be used during an exam or quiz.

Attendance:

- Arrive on time.
- Always whisper when the instructor is lecturing.
- If the student needs to leave early, the student must let the instructor know before the class starts, and choose a seat that minimizes disruption to the class when leaving.

- If the student has to miss the class, the student must send an email to let instructor know before class and explain the reason.
- If the student is sick and contagious, the student should not come to the class and risk getting others sick. Email the instructor before class to inform him of your decision.

Student responsibilities:

Each student is responsible for the successful completion and submission of all assignments and projects.

The instructor will not review your assignments or projects before grading for the entire class to ensure fairness. The instructor will, however, help you understand the expectations and clarify the requirements. Spot assessments will help you to outline questions and receive pre-due date feedback. **Whenever you have questions related to our studies**, please send an email to the faculty member, or attend his office hours, posted on page one, above.

The instructor will not debug assignments or projects for an individual student. The instructor will, however, help you gain knowledge and skills in analysis and design, problem solving, coding, testing and debugging, and answer specific questions about course topics. **It is always appropriate to discuss bugs and other complications with the instructor, however, the process of fixing problems rests with each student, exclusively.** Make sure you have spent significant time and effort in researching and working on your own before asking help. **To help you in this effort, many code examples are posted on our course GitHub, and to support your learning, each lecture is posted on our youtube channel. Be sure to watch each youtube instructional item at least once!**

Team Work

No team work will take place during 311 this term.

10. University Policies

Students with Disabilities: Upon identifying themselves to the instructor and the university, students with disabilities will receive reasonable accommodation for learning and evaluation. For more information, contact Services to Students with Disabilities at <http://dsa.csupomona.edu/drc/>.

Academic Integrity: Students should understand or seek clarification about expectations for academic integrity in this course (including no cheating, plagiarism, or inappropriate collaboration); neither give nor receive unauthorized aid on examinations or other course work that is used by the instructor as the basis of grading; take responsibility to monitor academic dishonesty in any form and to report it to the instructor or other appropriate official for action.

Cheating and Plagiarism: Cheating is the actual or attempted practice of fraudulent or deceptive acts for the purpose of improving one's grade or obtaining course credit; such acts also include assisting another student to do so. Typically, such acts occur in relation to examinations. However, it is the intent of this definition that the term 'cheating' not be limited to examination situations only, but that it include any and all actions by a student that are intended to gain an unearned academic advantage by fraudulent or deceptive means. Plagiarism is a specific form

of cheating which consists of the misuse of the published and/or unpublished works of others by misrepresenting the material (i.e., their intellectual property) so used as one's own work. Penalties for cheating and plagiarism range from a 0 or F on a particular assignment, through an F for the course, to expulsion from the university.

In assignments where code is required, students who allow their work to be copied will receive a zero on the first episode of cheating. Forewith, meetings with the university will follow, to escalate, potentially, to expulsion. All code you submit must be coded by you without copying from another source.

Computing Resources: At Cal Poly Pomona, computers and communications links to remote resources are recognized as being integral to the education and research experience. Every student is required to have his/her own laptop/computer or have other access to a computer with all the recommended software for this course. Find out more about how to access to the university's information resources from Information Technology Services.

Copyright Policy: Copyright laws and fair use policies protect the rights of those who have produced the material. The copy in this course has been provided for private study, scholarship, or research. Other uses may require permission from the copyright holder. The user of this work is responsible for adhering to copyright law of the U.S. (Title 17, U.S. Code). A full description of Cal Poly Pomona's copyright policy is included in the University's Intellectual Property policy. The course web site contains material protected by copyrights held by the instructor, other individuals or institutions. Such material is used for educational purposes in accord with copyright law and/or with permission given by the owners of the original material. Students may download one copy of the materials on any single computer for non-commercial, personal, or educational purposes only, provided that (1) do not modify it, (2) use it only for the duration of this course, and (3) include both this notice and any copyright notice originally included with the material. Beyond this use, no material from the course web site may be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way without the permission of the original copyright holder. The instructor assumes no responsibility for individuals who improperly use copyrighted material placed on the web site.

11. Tentative Course Schedule and Graded Items

Term Grading

The Term will contain 7 micro-projects, each worth 6% or 5%, and due at the close of each unit in the Lesson Plan. A Final term project will take place worth 20%.

Each week a quiz is given, based upon the content of the prior week. A total of 9 days of programming are allotted for each quiz. Each quiz is based upon the coding work assigned for each microproject, affording students the ability to account for their skills in either coding or during an administered exam. Code will be covered on most if not each question, each week. Therefore, performing the micro project is the best preparation for each quiz.

In cases where students receive a 50% or below on a quiz, the Instructor will view their blog, and apply the grading rubric, below to their micro project grade.

Weekly Quizzes	Unit	Weight (%)	Due
Quiz or micro project is graded each week.	1	6	6 / 29
	2	6	Jul 13
	3	6	July 20
	4	6	July 25
	5	6	August 1
	6	6	August 8
	7	6	August 15
extra credit microproject	8 *	6	August 24
total microproject points		42	
total Final Project points		8	August 31
Term value of final project		16%	
Total points		50	

* MP 8 is optional, and can be submitted to replace up to six points on prior MPs.

Term Final Project

Interactive Web Product. The final project compiles each micro-project into a cohesive interactive web product. Compile a product video on a promotional page of the product's website. ASP.NET is to be used. Students will integrate resources from Microsoft in creating their individual project. Resources for the project are located at this address, <https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/index> This link will also be posted to the course GitHub page. Students should plan to devote 15 hours to the project, minimum.

How to Submit Your Project for Grading

1. **Public blog.** Students will demonstrate their project on their term blog, where their micro projects are posted.
2. **Submission deliverable.**
 1. Students will copy/paste the code they used to produce the site in **a single .pdf document**.
 2. In this document, screen shots of each working UI are to be inserted, to accompany their code. The document should show the working product with the code they used to produce it.
 3. A dropbox will be sent to students on the due date, via CPP email, to assist with their submission.
 4. Only one submission per student may take place, and must be submitted by the closing time of the final exam, as posted on the school's final exam schedule (and on our GitHub, for convenience).
 5. Multiple documents, unorganized images, or departure from rule 1 results in a zero.
 6. Your submission's filename must contain your full name, and the words, '311 ASP project'.

Grading Rubric, for ASP.NET projects:

- A. Project completes an ASP.NET project with several key elements. Elements include
 1. integration of SQL Server
 2. deployment of a web container (IIS) on a personal device (laptop / PC)
 3. integration of SQL queries producing HTML markup
 4. integration of a responsive CSS in a uniform fashion on all pages,
 5. evidence of input / update and delete on database contents
 6. a basic security mechanism.
 7. use of URL parameters
 8. use of the ASP.NET model / view paradigm
- B. Project is missing 10% of the A criteria.
- C. Project is missing 20% of the A criteria.
- D. Project is missing 30% of the A criteria.
- E. Project is missing 40% of the A criteria.

Micro-Projects and Quizzes

Each week, a quiz is given, and is valued the same as a microproject. Students should prepare the micro project for the above due dates, then take the weekly quiz. You receive credit by submitting a blog post of your work, where a one-minute youtube video is present.

If a student receives a 50% or below on the weekly quiz, the Instructor will review their blog, which should contain a video of the microproject, due for that week. This way, if a student codes well but does not test well, their code will be reviewed. Also, in cases where students do not complete the microproject, their quiz grade will suffice.

For each unit in the schedule, below, a small project is made by the student to practice and implement the technologies taught in class. Helpful source code will be presented and made available online on our course GitHub. Students will be expected to show competency implementing the code demonstrated for the unit in a working micro-project.

Deliverable. The coding practices taught in the unit should be present on each page in some functional, explainable way. The student should show that they have learned how to use the skills, then apply them in a very simple demonstration. For each project, an outside observer should see them moving data within their page, then display it in HTML markup in a very simple business process.

Each student will use a public blogging service to describe their work in a brief video demonstration (youtube is adequate for this purpose). The assignment is a blog post, completed by the due date, which links to their functioning micro-project video. **Be sure, in each video, to show the code you used, using the Inspect Element feature of your browser, not in a separate IDE.**

Best Practices

For your micro-project, understand that A projects are small portfolio pieces, adequate to show employers a wide range of web development skills.

Due dates

Micro-projects must be submitted by end of the next unit. For example, class ends on a Thursday, then the micro-project will be at midnight. Your micro-project must be online and accessible for full credit.

Grading Rubric (for each weekly micro project)

A

Student accomplishes the week's micro project goals, per the table "Weekly micro project goals," on the following page.

B

Student accomplishes 85% of the coding goal

C

Student accomplishes 75% of the coding goal

D

Student accomplishes 65% of the coding goal

E

Student accomplishes 55% of the coding goal or below

Lesson Plan, Lecture Topics and Session Objectives

Months	Class sessions	Lecture Topic	Unit # on github
June	20	Basic Javascript. Basic user interface interactivity, HTML elements	1
	22	Responsive project element demonstration and discussion. Navigation, URLs and basic UI for HTML 5, using bootstrap CSS.	
	27	Javascript objects. Manipulating member variables and functions in javascript core language	2
	29	Responsive project element demonstration and discussion. Alerts, labeling, wells and other means of displaying data to users.	
July	6	Handling Data with Javascript Objects. Javascript Object Notation (JSON), java objects, arrays, array traversal, functions, forEach iteration, Document Object Model (DOM) and javascript, text nodes	3
	11	Responsive project element demonstration and discussion. List elements, tabular outputs, menus with dynamic content.	
	18	Cloud Security and Querying NoSql Databases. OAuth, Amazon Web Services SDK, Identity and Access Management (IAM), DynamoDB and DocumentClient, event-driven javascript, Web Identity Federation, Third Party Authentication Systems and Cloud Computing	4
	20	Responsive project element demonstration and discussion. The HTML div and CSS styles. CSS grids, page layout and row-based container elements in CSS	
	25	Integrating services into client-side javascript. HTML Elements integrated with cloud computing APIs	
	27	Responsive project element demonstration and discussion. Continued work in developing means to display interactive output from data, in rows	
August	1	Basic Get, Put, Update, Delete on NoSql cloud environments. Transferring cloud JSON into HTML elements	5
	3	Responsive project element demonstration and discussion. Triggering navigational elements via javascript and the window object in the HTML DOM	
	8	Persisting client-side data with cookies and adding interactivity to interactions on the web. URL parameters in the DOM	6

Months	Class sessions	Lecture Topic	Unit # on github
	10	Responsive project element demonstration and discussion. Integration of prior weeks' discussion into cookies. Leveraging data transfer via URL parameters	
	15	HTML FILE protocols and uploading images to the cloud	7
	17	Responsive project element demonstration and discussion. The responsive image, img tag, and the row of divs. Building interactivity to links which involve data transferred over URL parameters, including images.	
	22	Final Project work time. Get personal consultation, coaching and time to discuss project.	8
	24	Final Project work time. Get personal consultation, coaching and time to discuss project.	
	TBA	final exam, final project submission due date	

Weekly micro project goals

Microproject	Objective	Consult
1	Be sure to try examples 1 through 12 in our unit 1 .pdf tutorial on basic javascript in our github. Create four .html pages, where responsive design is taking place (bootstrap). Use a drop-down navigation menu on each page. Enable each page to contain text inputs and paragraph elements which respond to the user's input.	http://getbootstrap.com/getting-started/#examples-navbars and unit 1 code examples. Be sure to view our youtube, also.
2	Add one javascript object to each of your new pages. Each object should have a different name, but their data content may be different. When the visitor enters each page, they should be able to click a button which displays the object's data inside a bootstrap panel. Set the title and panel body using content from your object on each page.	http://getbootstrap.com/examples/theme/#
3	For every page where you implemented a javascript object in unit 2, add a JSON array of data, as we did in our samples. For those pages, implement a row of columns using bootstrap, then add div elements for each row of the data. Your data array in JSON should appear within several columns.	http://getbootstrap.com/examples/grid/

4	Begin with the pages you created for unit 3. Implement a query on pages where you handle JSON. Create one table in dynamoDB for each query (3 or 4, likely). Present the results of your queries in your bootstrap columns, as in unit 3. You may also produce more than one query on a page, associated with different buttons.	
5	Begin with the query results you got for each query in MP 4. For each div you create, with query results, add an HTML href, or link element. This element should contain results from your query, to construct a valid URL. Hence, you should create a column in your DynamoDB which contains a string sufficient to create a hyperlink. Then, when this column data loads within the page, a link is constructed, concatenating other URL strings. The link should be underlined. Create the link in a responsive way.	http://getbootstrap.com/examples/theme/#
6	Use your new skills with CRUD. Building upon our samples, develop three pages which maintain your authenticated connection to AWS. On these three pages enable your first page to create two values of data in your database. On the second page, enable those values to be updated, after a get. On the third page, enable for your data to be gotten, then deleted. Assume that you hard-code the itemId to fetch the appropriate item from the table in your javascript on pages two and three of your mp.	
7	Return to micro project 6. In the page that does a 'put' on your database, store the information the user inputs to a cookie. Then, in the 'update' page, change the information in the cookie. Demonstrate the use of URL parameters by sending the user to a new page, using <code>window.location</code> after the update is completed by AWS. When you send the user to the new page, after the update, practice using URL parameters by loading the URL in <code>window.location</code> with the variables you just saved to the cookie. Have the console show the URL parameters once you arrive in the new page.	use this reference to do <code>window.location</code> : https://appendto.com/2016/04/javascript-redirect-how-to-redirect-a-web-page-with-javascript/

8	<p>Extra Credit Micro-project. Integrate each prior micro project to produce a four page site.</p> <ul style="list-style-type: none"> • On this site, you should offer a login page, directing the user to Facebook. • After authenticating with Facebook, direct the user to a registration page, where they fill in information relating to their complete name, email, and uploading a picture. • When this information is submitted to the database, redirect them to a home page. • Enable a drop-down menu to contain a link to a profile page. • The profile should contain the latest inputs to their profile including their photograph, name and email. • A button will be placed on the profile page enabling them to edit their profile, which should update their profile, after getting its latest settings. • They should also be able to upload a new profile image. • An additional 'delete account' button will allow for the profile to be deleted in the database after an alert window asks them to confirm their intention. 	
---	--	--

Syllabus Version History

1.0 - 1.7. Added microprojects on a weekly basis, mp 1 through 7.

1.8. Added Extra credit microproject. Updated actual point values for assignments (adjusting each mp from 10 to 6 points except for ups 2 and 3, which stand at 5 points. Total course points adjusted from 90 to 48, maintaining the possibility for extra credit (6 points). Clarified grading for quizzes and microprojects.

1.9 Adjusted language to specify quiz, mp and grading rubric for summer '17 courses.

2.0 Added language on pages 2 and 3 to specify how to turn in microprojects via email, and an updated due date to do so