Syllabus for CIS 3110: Interactive Web Development

California State Polytechnic University, Pomona
Spring 2020

Instructor: Stefan Bund, MS
Office: Room 1011, Building 163, 1st floor conference room during office hours
E-mail: slbund at cpp.edu (replace at with @)
Course Website: https://github.com/stefanbund/311
Office Hours: Tuesday, Thursday 2 - 3:45 pm, office above
Class Location and Meeting Times: Building 98C Room 5 - 17

This syllabus governs all policy for the course. It is subject to change under certain conditions.

Guide to the Syllabus

Class Meetings and Office Hours, Holidays

Class Meetings and Office Hours, Holidays

Term Dates

Classes run each week from January 22st through May 9th, Tuesday and Thursdays.

Final Exam Time

TBA, currently on the university website. Final exam day will be May 14 or 16. Students will be informed of the final exam time via email, and an announcement via Blackboard.

Class Meeting Times and Locations

7:00 PM–8:15 PM TuTh, Building 9, room 241 (Engineering Building)

Holidays

Spring Break takes place April 30 through May 5th. Two normal class sessions will be missed during this week, yet assignments may be due, to be announced.

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, React.js, JQuery and client-facing user experiences. 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.

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.
- Collect, validate and process information entered by users via web forms.

Prerequisites

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

Textbook and Software

No required textbooks.

Required Web Development Software for assignments, one of the following:

- atom IDE, atom.io

Exams, Projects and Assignments

A detailed list of all graded assignments is provided in the
Tutoring
For free tutoring on campus, contact the CIS department in the CBA Administration Building.

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.’

Class Communication and Getting Help

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 3110" in the subject line, or it may not be read. Please consult the syllabus before sending emails, especially in the area of microprojects. **Messages sent through Blackboard will not be read.**

Coding Questions

In cases where you have a technical question, please post this to our Github at

- `https://github.com/stefanbund/311/issues`

You may also send the professor an email at `slbund at cpp dot edu`.

How to Write Emails

Expect that your email is read on a mobile device, to return a reply to you very quickly and build value in your experience. Thus, compose your email carefully:

1. include screenshots of errors in the Javascript console, in your Browser
2. run Inspect Element on the running page, screen shot the code you are running
3. error codes given by your IDE, browser are really helpful!
4. include only a short number of sentences, and try to limit the email to one or two precise questions. This guarantees a quick reply.
5. The closer you ask questions to a deadline, the harder it is to get a fast reply. Schedule your work so you get help efficiently.

General Course Expectations on Help and Technical Assistance

- Your process should include reading the syllabus, reading the assignment, reviewing our learning materials, trying code, collecting results, **then** asking about running code. This will dramatically speed-up your experience. I would prefer not to answer your email with `please read syllabus page 9 under heading....`
- Please understand that very general questions are harder to answer, so emails with the subject line `PLEASE HELP` (multiple emoji), or `I AM CLUELESS` (emoji 1 and 2) are harder to manage, and are likely to be delayed.

- Emails with precise subject lines like `DynamoDB Error in AWS`, or `Authentication Failure on Facebook` will be **potentially** answered quickly throughout the day, Monday through Friday.

Course materials

Lecture presentations, assignments, projects documents, classroom exercises and solutions, will be posted on Github. All graded assignments and projects will be visible in Blackboard Gradebook. Our github repository is:

- https://github.com/stefanbund/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.

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.

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 your 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!**

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 explusion. 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, re-published, 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.

Tentative Course Schedule and Graded Items

Course Projects

The course grade is calculated using the items, below.

Phased Project #	Weight (%)
1, due first week in October	10
2, due first week in November	10
final exam presentation, due during final exam	15
Total points	**25**

Grading Policies for Projects

1. For Project 1 and 2, projects submitted at the deadline are eligible for full credit. Projects may be submitted after the due date, but cannot qualify for scores higher than 85%.
2. Each project must demonstrate each of the skills taught during the project period, ie after the prior project, and before the due date.
3. The subject matter of the site is determined by the student. Concepts should address real-world demands, from an industry of interest to the student.

General Course Calendar

Graded Assignments:
monthly projects, due at the start of each respective month:
March 1
April 1
May 1
Final Project, due at final exam time

Topics	Lesson assets
html essentials: 1. core html, body, page, dom 2. head, meta, link ref, css 3. Paragraphs, ordered lists, underlines, bolds, bulleted lists 4. Links, navigation menus, download attributes, mail-to 5. Description lists, blockquotes, citations, quotations, inline citations, superscript, subscript, address, code, time/date markup 6. Document structure, site map, website information architecture 7. Debugging html, W3C validation	1. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Getting_started 2. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML 3. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/HTML_text_fundamentals 4. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Creating_hyperlinks 5. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Advanced_text_formatting 6. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure 7. https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Debugging_HTML
CSS essentials: 1. Fundamentals of css 2. Integrating with HTML. External style sheets, classes, location styling (li), state defining styles,	1. https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS 2. https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Getting_started 3. https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_is_structured 4. https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/How_CSS_works

selectors/combinators 3. Selectors application, properties, values, functions, @rules, shorthand, comments, whitespace 4. Theory: DOM and CSS, DOM node trees, browser as an interpreter/compiler 5. Customizing fonts, colors, borders, using MDN reference to all css properties	5. https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/Using_your_new_knowledge , https://developer.mozilla.org/en-US/docs/Web/CSS/Reference
Project 1: incorporate each skill into a five page site.	Due march 1

Javascript, Bootstrap, Applications

1. Javascript essentials: functions, inline vs external. createElement, appendChild, addEventListener, button driven events. Async and defer keywords, commenting. 2. Applying javascript in an interactive html game 3. Debugging and troubleshooting javascript. Syntax vs logic errors. Console, Inspector, in-browser tools vs IDE tools 4. Javascript variables: strings, number types, arrays, objects. Constants, var and let. 5. Arithmetic, comparison operators 6. String fundamentals, practical string methods (strings as objects) 7. Array programming fundamentals 8. Project: a story generator	1. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript 2. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/A_first_splash 3. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_went_wrong 4. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Variables 5. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Math 6. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Strings , https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Useful_string_methods 7. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Arrays 8. https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/Silly_story_generator
Responsive web design essentials: 1. Responsive concepts, **bootstrap** 2. Linking to bootstrap cdn, use of basic classes in a layout 3. Containers, grid, Bootstrap grid system classes 4. Typography, images 5. Tables, figures 6. Buttons 7. Example 1: the checkout UI 8. Example 2: using the grid 9. Example 3: starter template 10. Navbar kitchen sink	1. https://getbootstrap.com/docs/4.4/getting-started/introduction/ 2. Cdn example 3. https://getbootstrap.com/docs/4.4/layout/grid/ 4. https://getbootstrap.com/docs/4.4/content/typography/ 5. https://getbootstrap.com/docs/4.4/content/tables/ , https://getbootstrap.com/docs/4.4/content/figures/ 6. https://getbootstrap.com/docs/4.4/components/buttons/ 7. https://getbootstrap.com/docs/4.4/examples/checkout/ 8. https://getbootstrap.com/docs/4.4/examples/grid/ 9. https://getbootstrap.com/docs/4.4/examples/st

	arter-template/ 10. https://getbootstrap.com/docs/4.4/examples/navbars/
Project 2: five page site incorporating each skill	Due april 1

React.js Web Applications

Lesson 1: AWS EC2, Cloud 9, node.js and create-react-app setup	
Lesson 2: negotiating and understanding the create-react-app template system	
Lesson 3: dependencies and remaking the root App.js	
Lesson 4: adding new components, custom built	
Lesson 5: learning state	
Lesson 6: Lifecycle	
Lesson 7: Mapping data using inline functions, inside of render(), employing javascript methods inside of render()	
Lesson 8: props and passing data between components	
Lesson 9: component templating	
Lesson 10: implementing uniqueness among data	
Lesson 11: icons, dates via third party libraries	
Lesson 12: clicks, prop handling	
Lesson 13: forDisplay, passing visibility settings via props	

Lesson 14: Toggle elements in UI	
Lesson 15: capturing user input as state	
Lesson 16: adding to the master UI's state	
Lesson 17: initiating component #3, the search component	
Lesson 18: conveying state variables to the component, to facilitate a sorted search	
Lesson 18 part 2: fulfilling a complete set of sort operations, in UI	
Lesson 19: filtering data interactively in parent UI, where state is present	
Lesson 20: capturing UI input to drive search and filtering, on arrays	
Lesson 21: modifying array data, using UI inputs in React	
Project 3, demonstrate the ability to create a react.js application, customizing our demo version into five pages, with a bootstrap navigation bar on the first page (app.js)	Due may 15, at close of final exam period