C S 77A: ADVANCED WEB APPLICATION DEVELOPMENT

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Summer 2022
Units:	4.5
Hours:	4 lecture, 2 laboratory per week (72 total per quarter)
Advisory:	C S 22A, C S 30A, C S 40A, and GID 55.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- Ethically create data-driven web applications that work with client or server storage systems.
- Ethically create web pages using modern versions of Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript, and the Document Object Model (DOM), and demonstrate how they interact together within a web document using techniques that are responsive to differing screen sizes.
- Ethically create rich Web applications that deliver similar features and functions previously associated with desktop applications through the use of modern libraries or frameworks.

Description

Design and develop applications that deliver similar features and functions normally associated with desktop applications using modern web client and server technologies.

Course Objectives

The student will be able to:

1. Understand the history of the web, and use web tags, and Application Programming Interfaces (API).
2. Design, create, and organize modern HTML documents.
3. Construct basic web forms using HyperText Markup Language (HTML).
4. Embed audio and video in applications.
5. Use web API in rich internet applications.
6. Improve caching and storage for rich internet applications.
7. Use Cascading Style Sheets (CSS) to enhance and style rich internet applications.
8. Use modern HTML controls in applications.
9. Evaluate client/middleware/server development tools.
10. Create data-driven web applications.
11. Discuss and analyze professional ethics and societal power structures.
12. Use responsive web design for differing screen sizes.
13. Use security techniques.

Course Content

1. Explore web history, tags, and Application Programming Interfaces (API)
   a. History of HyperText Markup Languages (HTML)
   b. Modern HTML features
   c. Structural, content, and application-focused tags
2. Explore designing, creating, and structuring modern HTML documents
   a. Content models
   b. Understanding the outline algorithm
   c. The role of div tags
   d. Using ID and class attributes
   e. DOCTYPE declarations
   f. Character encoding
   g. Compatibility testing using browsers and mobile devices
   h. Structure of basic page, top level elements and interior content
   i. Building headers
   j. Checking document outlines and ensuring cross browser structure
3. Construct basic forms using HTML
   a. Modern input types
   b. Setting form autofocus
   c. Using placeholder data
   d. Marking required fields
   e. Working with number inputs
   f. Using date pickers
4. Embed audio and video in applications
   a. Adding audio
   b. Encoding audio
   c. Adding video
   d. Encoding video
5. Learn and apply usage of web API in rich internet applications
   a. Canvas API overview
   b. Adding canvas content
   c. Drawing in the canvas environment
   d. Drag-and-drop API overview
   e. REpresentation State Transfer (REST)ful API and Create, Read, Update, and Delete (CRUD) operations overview
6. Improve caching and storage for rich internet applications
   a. Offline applications overview
   b. Geolocation API overview
   c. Web storage API overview
7. Demonstrate usage of Cascading Style Sheets (CSS) to enhance and style rich internet applications
   a. Modern CSS overview
   b. Enhancing typography
   c. Using @font-face
   d. Styling modern HTML with modern CSS
   e. Using CSS transitions
8. Demonstrate usage of modern HTML controls in applications

   a. Email address input
   b. URL input
   c. Telephone number input
   d. Search field input
   e. Datalist form control
   f. Slider form control
   g. Spinner form control
   h. Calendar form control
   i. Color form control
9. Evaluate client/middleware/server development tools
   a. Tradeoff analysis some of the current languages, tools, frameworks, and/or libraries
10. Create data-driven web applications
   a. Use client and/or server storage systems
11. Discuss and analyze professional ethics and societal power structures
   a. Ethical and societal topics and issues that arise in the news
   b. Nuclear war historical effects on internet infrastructure design and implications for web and cloud services
   c. Professional ethics codes and laws
   d. Ethical implications of computer hardware production, reusing, recycling, and disposal
   e. Analyze how software developers contribute to, resist, or otherwise intersect with structures of inequality and hierarchy in societies
   f. Societal implications of different types of software producing organizations (such as not for profits, for profits, non-profits, worker cooperatives, customer cooperatives, benefit corporations, B corporations, etc.)
   g. Computer related industries and customer capture economic models
   h. Unionization in technology companies and organizations
   i. Designing web applications with low and sustainable environmental footprints
   j. Societal implications of software licenses and terms of service
   k. Power of web-based computing to transform society
   l. Web application design to support privacy
   m. Data ethics and data stewardship
   n. Digital Rights Management in web browsers
   o. Net Neutrality and the internet as a ubiquitous public utility
   p. Societal need and technological support for “Do Not Track” Global Privacy Control
12. Use responsive web design for differing screen sizes
   a. CSS media queries
   b. Flexible images and media elements
   c. Flexible grid
13. Use security techniques
   a. SSL/TLS, HTTPS, SSH, SFTP
   b. Sessions, cookies, and web storage API
   c. Single sign on (such as via OAuth)

Lab Content

The following are the general lab topics that must be covered. Any following lab topic may be separated and/or combined with any other lab topic(s).

1. Semantic web
   a. Modern HyperText Markup Language (HTML) documents
   b. Basic forms using HTML
2. Web/Rich internet applications
   a. Tags and Application Programming Interfaces (API) to build web/rich internet applications
   b. Modern Cascading Style Sheets (CSS) to enhance and style web/rich internet applications
3. Front end and media technology in web applications
   a. Audio and video media
   b. 2-D and/or 3-D web API(s)
   c. Widgets and/or animation/effects
4. Middleware and server technology
   a. Web servers and data servers
   b. APIs and controllers
5. Front and back end data storage and modeling for web/rich internet applications
   a. Databases
   b. Caching and offline storage
6. Native apps
   a. Mobile apps
   b. Desktop apps

Labs will typically be structured as follows:

1. Read and run the code that utilizes the associated lab topic(s)
2. Create a web application using the associated lab topic(s)
3. Discuss design and implementation tradeoffs of related techniques and tools

Special Facilities and/or Equipment

1. Access to a computer laboratory with web browsers, web development software, web server and middleware software, and database software. Computer laboratory can be provided as a web-based and/or virtualized online service(s).
2. A website or course management system with an assignment posting component (through which all lab assignments are to be submitted) and a forum component (where students can discuss course material and receive help from the instructor). This applies to all sections, including on-campus (i.e., face-to-face) offerings.
3. When taught via the internet, the college will provide a fully functional and maintained course management system through which the instructor and students can interact.
4. When taught via the internet, students must have currently existing email accounts and ongoing access to computers with internet capabilities.

Method(s) of Evaluation

Formative exercises, projects, and quizzes requiring students to write code applying covered technology topics
Formative exercises, discussion forums, projects, papers, and/or quizzes regarding covered ethics and societal power topics

Final examination requiring students to present projects applying topics covered in the lectures, reading, and programming assignments
Evaluation of programming assignments based on correctness, documentation, code quality, and test plan executions

Method(s) of Instruction

Blended instruction including discussion of topics
Online labs (for all sections, including those meeting face-to-face/on campus) consisting of:
1. An assignment webpage located on a college-hosted course management system or other department-approved internet environment. Here, the students will review the specification of each assignment and submit their completed lab work
2. A discussion webpage located on a college-hosted course management system or other department-approved internet environment. Here, students can request assistance from the instructor and interact publicly with other class members
3. Collaborative team projects
When course is taught fully online:
1. Instructor-authored lecture materials, handouts, syllabus, assignments, tests, and other relevant course material will be delivered through a college-hosted course management system or other department-approved internet environment

Representative Text(s) and Other Materials

Benjamin, Ruha. Race After Technology: Abolitionist Tools for the New Jim Code. 2019.

Boduch, Adam, and Roy Derks. React and React Native: A Complete Hands-on Guide to Modern Web and Mobile Development with React.js, 3rd ed.. 2020.

Robbins, Jennifer. Learning Web Design, 5th ed.. 2018.

Subramanian, Vasan. Pro Mern Stack: Full Stack Web App Development with Mongo, Express, React, and Node. 2019.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

1. Reading
    a. Textbook assigned reading averaging 30 pages per week
    b. Reading the supplied handouts and modules averaging 10 pages per week
    c. Reading online resources as directed by instructor though links pertinent to software engineering
    d. Reading library and reference material directed by instructor through course handouts
2. Writing
    a. Writing technical prose documentation that supports and describes the programs that are submitted for grades

Discipline(s)

Computer Science