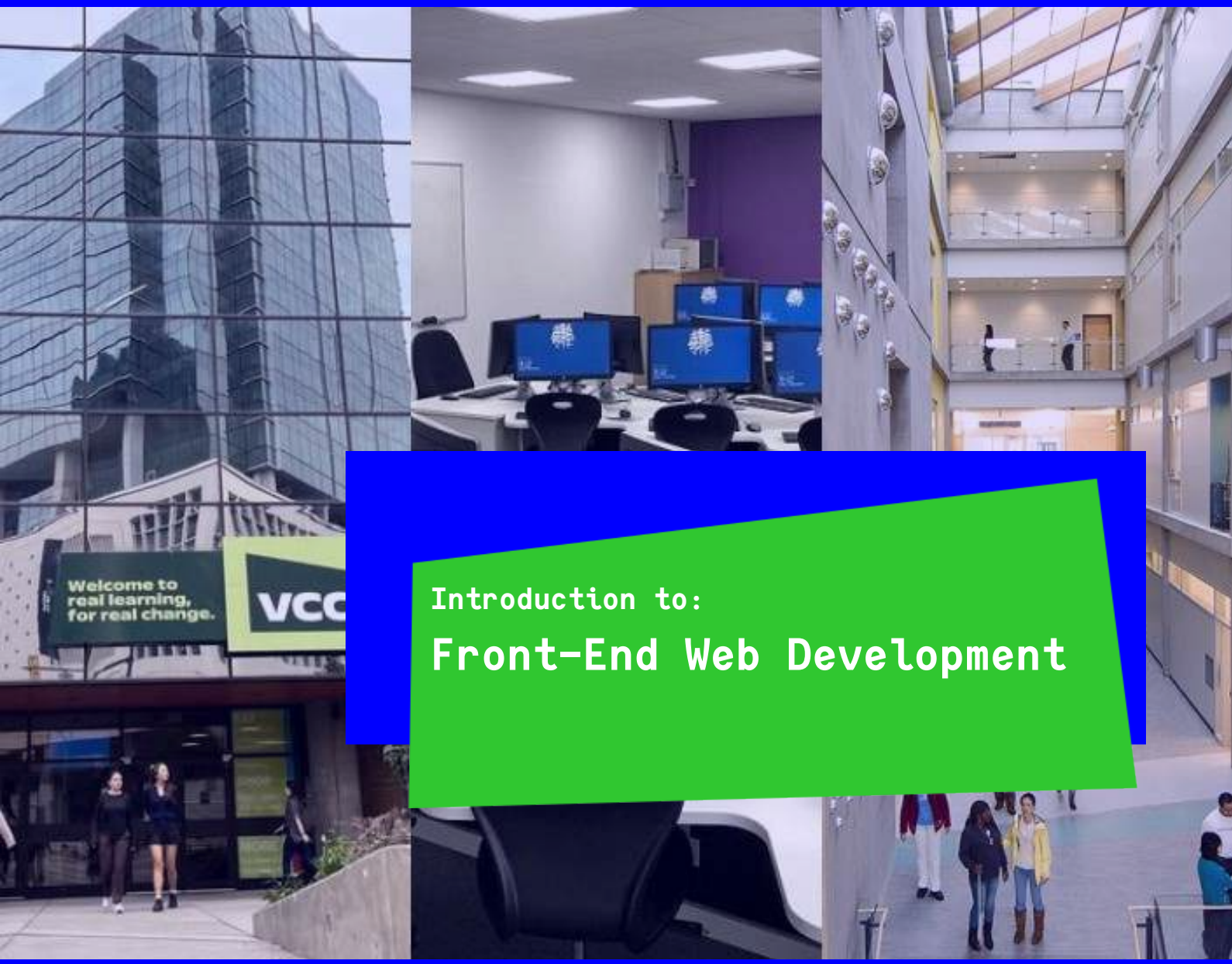
PIDP 3210: Curriculum Development
COURSE PROFILE

Instruct



Introduction to:
Front-End Web Development

Course Developer: Yuri Tricys
Date: March 21st, 2025

Stakeholders

In the context of a web development course, stakeholders can include a variety of individuals and groups who have an interest in or influence over the course's success. These stakeholders are crucial for the course's development, execution, and impact.

Here are some possible stakeholders:

- Course participants
- Educational institutions
- Course administrators
- Instructors and facilitators
- Technical support teams
- Software vendors
- Employers and hiring managers

While the fundamentals of web development are consistent across markets, because they align with standards regulated by institutions that are intended to be representative of a generalized and diverse body of stakeholders - institutions like: The World Wide Web Consortium [W3C], ECMA International, and the Web Hypertext Application Technology Working Group [WHATWG] - the way web technology frameworks and languages are taught in institutions can vary widely from institution to institution.

From the perspective of a curriculum writer, the primary stakeholder is, therefore, the educational institution. Any given institution that regularly sells web development courses, or has the intention of selling web development courses, should have some idea of what frameworks and technologies they'd like to see emphasized.

An institution that places graduates in financial firms may want courses that teach python as a backend technology, or even C++, which, from the front-end development instructor's perspective, might mean including web assembly in the curriculum, to give students an understanding of browser technologies they should be at-least aware of.

Having said that, JavaScript is the dominant front-end development language and web developers should have a strong understanding of vanilla JavaScript in order to learn and use any web development framework or technologies, such as WordPress, Node, Next.js or React.

While it is not necessary to consult stakeholders for the development of the initial course curriculum, profile, and outline, it will be necessary to consult educational institutions that intend to use these documents in their curriculum. In this way, the material can be modified to fit with institutional specific mandates, needs, and values.

Demand

According to Work BC, there are currently 7,395 workers employed in the web developers and programmers category (NOC 21234), 58% of whom are employed full-time.

The estimated median employment income, based on the 2024 job bank, is $78,085, which is an attractive enough salary to position web development as an in-demand career.

The Work BC forecasted openings, of 4,810 openings between 2024 and 2034, supports the popularity and attractiveness of web development as a career path.

[Source: https://www.workbc.ca/career-profiles/web-developers-and-programmers#career_overview]

Moreover, web development itself is a fast-changing, wide field, in terms of expertise expectations. The amount of material web developers need to know to operate optimally is immense, and that material continues to rapidly evolve. Continuous training is not just 'advantageous' to work successfully as a web developer, it is a mandatory part of the job description.

Web development is also a highly difficult job, in terms of the physical and mental demands. It comes with huge amounts of screen time, and can be overwhelming in terms of time commitments for many job holders.

On top of this, the average age of a working software developer (computer programming or web developer) is 28. Older developers typically move into other careers, including management, consulting, and teaching.

As a result of factors mentioned above, there is a continuous and steady supply of young adult web development students in the market of any given major metropolis. Such a market exists in the greater Vancouver area, the target market for this course offering, where it is represented in a growing supply of web development course offerings, and high-demand for web development instructors.

Program

This course is intended to be the first of two courses, both 16 weeks in duration. The path offered by the two courses combined is intended to provide a strong foundation of concepts to prepare students for higher level, more technology specific courses along the web development career path, such as web development and artificial intelligence, back-end web development in PHP, back-end web development in NODE, Advanced Web Design, Developing Websites in WordPress, and developing Progressive Web Application using Next.js.

While the content of the outline itself is derived from a blend of a competency based approach (OBE) and an outcomes based approach (CBE), only the outcomes based approach was used in the documentation, a Dacum was not necessary for curriculum design.

The competency based approach is represented in the capstone project, which is the creation of a one or more page website. The formal expectations of the course, however, are outcomes based, in order to provide a clear framework for what students should know and be able to do by the end of the course. The blended theoretical framework was chosen to help ensure course content is aligned with industry needs, and that students are prepared for the challenges they will face in their careers

A large proportion of students are anticipated to be in the young adult stage. Since selection of the course is voluntary, students are expected to have some background in computers, computer science, or computer programming. However, since many students are expected to be at the beginning of their web development career path, the material is targeted to those with no background in web development.

While the end-goal of the course is likely to be completed by students within the 16 week time frame, it is anticipated that some concepts will be missed or misunderstood by students.

In fact, this course, which is the first of two courses designed by the author and meant to be taken consecutively, could be easily split into two courses, and the two original courses together split into four. The pace is intended to be quick to expose students to as much material as possible in the given time frame under the expectation students will learn more in more narrowly focused second and third level courses.

JavaScript, for example, is a large subject that can be taught to varying degrees of expertise across many courses. While students of the first level of this course will be exposed to JavaScript, the second course offers a more thorough exposure, and a third JavaScript course will be beneficial for most students participating in both the first and second levels of this course.

The course schedule, then, is meant to be adjustable and iterative as live feedback is processed and applied during the instruction phase.

It should be noted, that individual lesson plans are a key to ensuring the teaching and learning strategies applied during the implementation of the course stay aligned with the learning goals and objectives documented in the course profile and outline.

Lesson plans that follow the BOPPPPS format, the Seven Step Lesson Plan Model , or the CARD approach are more likely to accurately cover the material presented in the outline and ensure course delivery aligns with learning goals and objectives.

It's the curriculum writer's intention to write each lesson plan for the two course series to ensure that lesson plan content aligns with the material in the outline, as closely as possible – after accounting for anticipated stakeholder adjustments – and that the lesson plans align with the learning goals and objectives expressed throughout these documents.

Assessments

Students will be quizzed every following week on the material they were presented in the previous week. During weeks students are not quizzed, they can work on take-home assignments. Some quizzes will be presented as live coding challenges.

Take-home assignments will be assigned on the first class of any given week and due at the end of the following week. Students will have two weeks to complete take home assignments.

There will be one lab during which students will interactively design and configure HTML forms.

Examples of assignments include writing a website plan, completing a wire-frame or prototype, and building a grid structure.

The Capstone Project, which is a website, will be graded against criteria presented to students in advance.

Outcomes Guide

Course Themes

The general theme for this course is that students will learn foundational principles of web development and web design while designing, coding, and publishing a single or multiple page website.

The instructor will act as a guide presenting students with knowledge in an interactive lecture style interspersed with interactive live coding activities and problem solving challenges.

Students will construct their own web development knowledge and experience through hands on activities, assignments, and projects.

The capstone project will be introduced from the start of the course and referred to again and again throughout the course.

Since web development in professional practice is largely a matter of applying theory to specific frameworks and libraries, such as WordPress, Laravel, Node, Next.js, Hugo, Shopify, etc ... students will be given the option of selecting any framework with which to build their website.

However, the class will be taught with a standard selection of NPM package manager libraries and a demonstration of how to use the Hugo static website generator to build a one page website.

Hugo is a good technology to use for an introductory course because it is easy to set-up and relatively easy to operate. However, there will not be much Hugo code in the course.

Since the final project is a one page website, the students will not be required to build their project in Hugo. The example project and lessons will be built in HTML, CSS, and JavaScript.

Outcomes Guide Flow Chart

Course Prerequisites/Entry Requirements

While a formal degree in computer science or a related field is not strictly necessary, having an advanced knowledge of one's computer system will be necessary to participate in the course. In addition, prior knowledge of coding will be helpful in constructing foundational web development skills and knowledge.

Prerequisites

- Must be able to install or remove programs on a laptop with either a Windows, Mac, or Linux operating system
- Should have a basic understanding of shell operations (enough to navigate to a directory, create a folder, create file, delete a folder, delete a file)
- Should be prepared to install and use a text editor such as VS Code
- Should be prepared to install such programs as XAMPP, WAMPP, LAMPP, and Node, as well as to use the NPM package manager
- Should be prepared to install, learn, and use GIT
- English 11 with a C- or equivalent

Recommended Characteristics

- Ability to communicate effectively in oral and written English
- Punctual and responsible
- Clean and well organized
- Ability to follow oral and written English instructions
- Good analytical skills

- Attention to detail
- Good technical aptitude

Concepts/Theories/Problems Scenarios

What Learners Will Understand

1. Historical perspectives on web development (why we have websites, who invented the DOM and CSSOM)
2. Beginning level understanding of HTML
3. Beginning level understanding of CSS
4. Beginning level understanding of JavaScript
5. Beginning level understanding of Web Design
6. Beginning level understanding of GIT (or subversion, if the student already uses that)
7. Beginning level understanding of Responsive Web Develoment
8. Beginning level understanding of the package manager NPM
9. Beginning level understanding of basic web security practices
10. How to collaborate and communicate effectively [workshop]

Skills / Practical / Hands on

What Learners Will Be Able to Do

1. Select and operate a text editor to write code
2. Correctly install, configure and use XAMPP, LAMPP, or WAMPP, or an npm package server (https-localhost, or) and run a local server
3. Back up code using versioning software (GIT)
4. Develop a plan to create a one or more page website
5. Create wireframe or prototype of a one or more page website
6. Create a valid HTML document that includes the mandatory HTML tags and attributes as specified in the course syllabus
7. Create a valid CSS document that adequately styles their webpage(s) in accordance with the styling specified in their wireframe or prototype
8. Create an HTML contact form from scratch (connected to email using a free service such as Zapier, or Google forms)
9. Create drop-down menu that opens and closes when a button is clicked using either vanilla JavaScript or a plug-in (if they are using a framework such as WordPress)
10. Run a lighthouse audit using Google Chrome to determine the performance efficiency of their website
11. Create CSS media queries to ensure their website is visible and functional on mobile devices, tablets, and desktop devices
12. Load and use a custom font on their website
13. Deploy a website to a production server (Github)

Assessment Activities and Tasks

How Learners Will Show Proficiency in the Course

1. Complete a wireframe
2. Demonstrate understanding of or build a design prototype
3. Build a webpage menu
4. Build a webpage navigation section
5. Build a webpage footer
6. Build a main section of a webpage
7. Use Lorem Ipsum to simulate content
8. Create content for a webpage
9. Create a design system
10. Build an interactive website form
11. Create a toggle for a menu in JavaScript
12. Connect a form to a data base or referral service [Zapier] using JavaScript
13. Trigger a form submission button using JavaScript
14. Create a CSS document for a website
15. Generate favicons for a website
16. Demonstrate an understanding of website image compression technologies/process
17. Demonstrate the use of an NPM scripting build pipeline
18. Demonstrate the use of a localhost server
19. Write a CSP level 1
20. Complete a lighthouse Audit for a webpage
21. Demonstrate an understanding of Chrome Dev tools
22. Demonstrate an understanding of a JavaScript loop
23. Demonstrate the use of JavaScript conditionals
24. Demonstrate the use of JavaScript Event Listeners
25. Demonstrate the use of Navigating the Document Object Model in JavaScript
26. Demonstrate an understanding of color spaces
27. Demonstrate the conversion of HEX Colors to RGBA, and HSL colors
28. Demonstrate the use of media queries to give a webpage responsive capabilities
29. Write a schema for a webpage
30. Demonstrate the use of source tags in picture elements to serve custom images for different screen sizes
31. Demonstrate the use of GIT versioning software to store and retrieve files in a root repository
32. Create a one or more page website and deploy it to a live server [Github]

Course Outcomes

What Learners Will Know After the Course

1. Demonstrate a beginning level understanding of HTML and CSS
2. Demonstrate a beginning level understanding of design
3. Demonstrate JavaScript proficiency at the beginners level
4. Demonstrate Responsive Design
5. Demonstrate proficiency of the version control system GIT, at the beginners level
6. Build A Simple Web Application
7. Publish A Simple Web Application
8. Implement Basic Web Security Practices

Summary

In summary, this profile represents the core objectives and outcomes anticipated from a course offering for beginner web development students.

Stakeholders for the course will not be consulted until the course is under the consideration of an educational institution, at which point the course profile and other documentation, such as the course outline and completed individual lesson plans, will be amended to align with the expectations of the institution.

There is a significant and ongoing demand for web developers – and therefore web development courses – due largely to the fast-changing field of web development technologies, high requirements for expertise in the field of web development, and quick turn-over of web developers into more senior non-coding roles.

This competency and outcomes based course is designed to be the first of two courses, both 16 weeks in duration, that will be provide students with a strong foundation of concepts and skills to prepare them for higher level web development courses.

Students will be assessed on a variety of quizzes, take-home assignments, live coding challenges, labs, and a capstone project.

Students will be expected to have a strong command of their computers in order to keep pace with the course, however, the final project will be designed to be completed by beginner level students.

The course, however, should be considered difficult.