PIDP 3210: Curriculum Development
COURSE OUTLINE

Instruct

Introduction to:
Front-End Web Development 1

Course Developer: Yuri Tricys
Date: April 11th, 2025

Course Description

The Introduction to Web Development course is designed to equip students with the foundational skills necessary to build and maintain websites.

This course covers the basics of HTML, CSS, JavaScript, and Design which are essential for creating the structure, design, and interactivity of web pages.

Students will learn how to write clean and semantic HTML to create the backbone of a website, including elements such as headings, paragraphs, images, and links. They will also delve into CSS to style their HTML, mastering the use of selectors, properties, and values to control the layout, colors, and fonts of their web pages.

Moreover, the course will introduce students to JavaScript, a programming language that adds dynamic and interactive elements to websites. Students will learn how to use JavaScript to manipulate the Document Object Model (DOM) and handle events.

Additionally, the course will cover responsive web design principles, ensuring that websites are accessible and functional on various devices and screen sizes.

Beyond technical skills, the course will emphasize problem-solving and critical thinking.

Students will engage in hands-on projects and assignments to apply their knowledge in real-world scenarios. They will learn how to debug code, optimize performance, and ensure cross-browser compatibility. The course will also introduce version control systems like Git, which are crucial for collaborative web development projects.

By the end of the course, students will have the ability to create and deploy a fully functional single page website, complete with a user-friendly interface and engaging content.

The curriculum is structured to be accessible to beginners, with no prior knowledge required. However, having a basic understanding of HTML, CSS, and JavaScript can help students grasp concepts more quickly.

Learning Goals

Upon completion of the course, students who have successfully fulfilled the course requirements will:

Understand the Basics of Web Development

Demonstrate a solid understanding of the fundamental concepts and principles of web development, including the structure and function of websites.

Demonstrate a beginning level understanding of HTML and CSS

Be proficient enough in writing clean, semantic HTML to structure web pages and using CSS to style and layout content effectively.

Demonstrate a beginning level understanding of design

Be proficient enough in beginning website design to create a wire-frame or prototype for a one or more page website

Demonstrate JavaScript proficiency at the beginning level

Acquire enough demonstrable basic knowledge of JavaScript, including variables, data types, control structures, functions, and DOM manipulation, to create a drop-down toggle for a menu

Demonstrate Responsive Design

Be able to create a responsive web design that adapts to different screen sizes and devices, ensuring a consistent user experience across platforms.

Demonstrate proficiency of the version control system GIT, at the beginning level

Be proficient enough with GIT to initialize repositories, commit changes, pull and push changes, and collaborate with others.

Build A Simple Web Application:

Demonstrate the ability to create a simple one or more page website using vanilla front-end technologies such as HTML, CSS, JavaScript, and NPM packages.

Note: The class will be taught in HTML, CSS, vanilla JavaScript, Hugo, NPM libraries, with a class overview of WordPress

However, students may choose to use any web development framework, such as those listed below.

Frameworks Not Taught in Class:

- WordPress
- Laravel
- Node
- React
- Vue
- Angular
- Gatsby
- Astro
- Squarespace
- Shopify
- Webflow

Publish A Simple Web Application

A website will be published to a live production server. Github pages will be used in the class; however, students may deploy to any server they choose. Note: SQL, MySQL, and phpMyAdmin will not be taught in the class.

Implement Basic Web Security Practices:

Understand and apply basic security practices to protect web applications from common vulnerabilities like cross-site scripting (XSS) and SQL injection.

Collaborate and Communicate Effectively:

Develop strong teamwork and communication skills, including the ability to work with designers, back-end developers, and project managers to deliver high-quality web projects.

Learning Outcomes

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]

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 mandetory 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 dropdown menu that opens and closes when a button is clicked using either vanilla javascript or a plugin (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)

Course Pre-requisites

While a formal degree in computer science or a related field is not strictly necessary, having an advanced knowledge of ones computer system will be necessary to participate in the course. In addition, prior knowledge of coding will be helpful for laying the groundwork of web development.

- 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

Assessment Methods

- Graded Assignments
- Quizzes
- In class problem solving sessions
- 1 Lab
- A Capstone Web Development Project
- A Final Exam

Instructional Methods

- Lectures, discussions, and audio-visual presentations
- Hands-on web development labs
- Group work

Instructional Strategies

The courses will take place in a classroom. There will be a range of activities and methods used to facilitate learning. The activities and methods will include instructor-led lectures, discussions, and audio-visual presentations, as well as hands-on live code demonstrations and problem solving exercises that students can participate in.

Some projects will be done collaboratively in 'hacker-meet' style. For example, groups will create a website form together from a provided template, which they can then adapt to use for their own website.

All classroom activities will take place in an authentic workplace environment providing students valuable experience for a successful entry into the job market as an web developer.

Course Format: In-person (VCC Downtown Campus)

Duration

16 Weeks (48 hours)

Class Hours per Week

3 weekly classes of 1 hr each

Textbook

To be announced

Supplies & Equipment

- Students will require a laptop with at least 1 GHz processor (recommend no less than 2.5Ghz), 2 GB RAM (recommend more than 4 GB), and at least 20 GB of hardrive storage

- Students will require a mouse, and keyboard (or laptop keyboard)
- Any operating system will do, Linux, Windows, or Mac

Topic Schedule

WEEK 1: Introduction to the web and websites - Setting up a dev shop

Class	Details
Class 1	Course Introduction [easy class]
Class 2	Introduction to the internet and servers [simple overview]
Class 3	Setting up a Dev Shop [Tooling]
Project/Take Home Assignment	Project: Configure text editor with localhost server, recommended plugins, install GIT
Text Book	Handouts [Text-book TBA]

WEEK 2: Introduction to HTML, CSS, & Design + First Web Page

Class	Details
Class 3	Introducing HTML
Class 4	Introducing CSS
Class 5	Introductory Design Concepts

Class	Details
Project/Take Home Assignment	Project: First Web Page [Minimal]
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 1

WEEK 3: HTML Level 1

Class	Details
Class 7	Review of Introduction - Relevant Concepts
Class 8	50 HTML Tags
Class 9	Introduction to GIT [Backup first web page to repo]
Project/Take Home Assignment	Take home Assignment: Clone repo, answer questions about content
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 2

WEEK 4: CSS Level 1

Class	Details
Class 10	Review of Introduction, Concepts: [Modules/functions/properties/@-rules/Psuedo-Elements/Types]
Class 11	CSS Properties
Class 12	CSS Combinators
Project/Take Home Assignment	Take home Assignment: add Grid, add flexbox to first webpage
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 3

WEEK 5: Design Level 1

Class	Details
Class 13	Introductory Design Concepts
Class 14	Introducing Figma
Class 15	Introducing Inkscape/GIMP mention
Project/Take Home Assignment	Take home Assignment: Wireframe or Prototype of single page website
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 4

WEEK 6: Hugo & Planning A Website

Class	Details
Class 16	Introducing Web Frameworks, CMS, Static Generators, Web Creation Technologies
Class 17	Hugo Themes
Class 18	Website Plan
Project/Take Home Assignment	Take home Assignment: Website Plan Template
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 5

WEEK 7: Introduction to JavaScript Level 1

Class	Details
Class 19	Getting JavaScript into a Website
Class 20	Syntax, Naming conventions, Literals
Class 21	Constants, Variables, Scope, Arithmatic operators, Expresssions
Project/Take Home Assignment	Take home Reading: [no assignment]
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 6

WEEK 8: Introduction to JavaScript Level 2

Class	Details
Class 22	Data Types, Scope [very quickly also Objects and Arrays]
Class 23	Window Object [foreshadow Prototype Objects]
Class 24	Events, and Changing the Window Object
Project/Take Home Assignment	Take home Assignment: Clone a repo, complete the tasks, submit
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 7

WEEK 9: Introduction to JavaScript Level 3

Class	Details
Class 25	Functions [also execution context, foreshadow]
Class 26	Conditional logic [also switches]
Class 27	Loops [for loop, for of loop, while loop foreshadow forEach, foreshadow iterators]
Project/Take Home Assignment	Take home Assignment: Clone a repo, complete the tasks, submit
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 8

WEEK 10: NPM Packages - Build Pipelines

Class	Details
Class 28	NPM - Manage/run packages in Hugo repo
Class 29	package.json / Writing Scripts for a package manager
Class 30	Using programs to compress images for different screen sizes

Class	Details
Project/Take Home Assignment	Take home Assignment: Add images to your website
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 9

WEEK 11: Introduction to Responsive Design

Class	Details
Class 31	Device Types, Screen Resolutions, Responsive Problems
Class 32	Targeting Screen Sizes with Media Queries
Class 33	Layouts, Fonts, and Images across Screen Sizes
Project/Take Home Assignment	Take home Assignment: Write media queries for your webpage
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 10

WEEK 12: Introduction to SEO

Class	Details
Class 34	On page vs off page SEO / Getting Search Engines to Notice You
Class 35	Key words, Descriptions, Titles, Meta Tags
Class 36	Creating a Schema
Project/Take Home Assignment	Take home Assignment: Write a Schema for you website
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 11

WEEK 13: Put Everything in the Head Tag

Class	Details
Class 37	What goes in here: Favicons, Color Themes, CSP, Etc.. [Generate Favicons]
Class 38	Meta Tags [Open Graph Protocol]
Class 39	Creating a Content Security Policy [CSP Level 1]
Project/Take Home Assignment	Take home Assignment: Introduction to JavaScript Modules (not tested)
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 12

WEEK 14: Introduction to Web Development Frameworks

Class	Details
Class 40	Overview of Frameworks and What they all have in common
Class 41	WordPress
Class 42	Node
Project/Take Home Assignment	Take home Assignment: Work on project/check project against handout
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 13

WEEK 15: Miscellanous Topics

Class	Details
Class 43	Typography
Class 44	Animations
Class 45	Workshop/Lab [Create a Form in Groups]
Project/Take Home Assignment	Take home Assignment: Assimulate Form Project with your own website
Text Book	Handouts [Text-book TBA] / Online Quiz: Week 14

WEEK 16: Exam Week	
Class	Details
Class 46	Audits & Performance
Class 47	List of Topics Not Covered [Foreshadow Next Course in Series]
Class 48	Exam: Mostly terminology, Some spot the code problems, Some match code to description, 1 hard abstract problem
Project/Take Home Assignment	No Takehome Assigment
Text Book	Handouts [Text-book TBA] / No Quiz, Capstone Project is due

Education and Education Support Policies

The program will be subject to the policies of the procuring institution. Those policies will be clearly published in curriculum documentation used to adminstered the course, as well as the documentation that is provided to students at the begining of the course and on the online administrative interface.

Grading System

Letter Grade (A-F), **Passing Grade: C**

Letter	Range
A+	100.00% - 90.00%
A	89.99% - 85.00%
A-	84.99% - 80.00%
B+	79.99% - 76.00%
B	75.99% - 72.00%

Letter	Range
B-	71.99% - 68.00%
C+	67.99% - 64.00%
C	63.99% - 60.00%
C-	59.99% - 55.00%
D	54.99% - 50.00%
F	49.99% - 0.00%

Evaluation Plan:

Type	Percentage
Attendance	05%
Take home Assignments	20%
Weekly Quizzes/Projects	20%
Lab	05%
Capstone Project	30%
Final Exam	20%

Course Outline Rationale

For this outline I used the template provided as an example for the PIDP 3210 course. From what I understand, after exploring syllabus templates provided by various academic institutions, the permanent record format is adequate for the purpose of this exercise, in that it serves as a guideline for the course and communicates necessary information to any parties that might be interested in reading the course outline, like course goals and objectives.

If necessary in the future, I can always adapt the outline to fit into any given institutional format, for example a syllabus, brochure, flyer, or permanent record.

While the content of the outline itself is derived from a blend of a compentancy based approach (OBE) and an outcomes based approach (CBE), only the outcomes based approach was used in the documentation, a Dacum was never created for the course.

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

While web development courses in general are typically behaviorist and given in both lecture and workshop styles, largely because the complexity of the material favors demonstration, in this course the intended relationship between learner and instructor is more cognitive and constructivist. Learners will be tasked with exploring many aspects of the material on their own while the instructor acts as a guide.

Any readers may want to note that the content schedule (topic schedule) is to be read as a draft. While the material covered is the material the author considers most advantageous for the students, the pace is probably overly ambitious, especially when considering the prerequisites.

While the end-goal of the course is likely to be completed by students within the 16 week time frame, it is anticipated 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 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 available time, under the expectation students will dive more deeply into material during 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.