**Intro to Web Applications - Course Schedule**

**Fall 2025 | Tuesdays & Thursdays, 1.5 hours each** **August 19 - December 2, 2025**

**Week 1: Course Introduction & Setup**

**August 19 (Tue) - Session 1**

* **PRE-CLASS VIDEO**: "Artificiality of Academic Writing" (15 min)
* Course overview and expectations
* What is web development? Frontend vs. backend
* Introduction to the web: browsers, servers, domains
* **Lab**: Account setup (GitHub, create first repository)

**August 21 (Thu) - Session 2**

* **PRE-CLASS VIDEOS**: "Reading Gravity" (12 min) + "Writing for the Web" (8 min) + "CRAP Principles" (20 min)
* Installing and configuring VS Code
* VS Code extensions for web development
* File organization and project structure
* **Lab**: Create first HTML file, basic VS Code workflow
* **Threaded Discussion 1 Launch**: Analyze and discuss 3 websites you use regularly - what works well and what doesn't? Consider accessibility and usability.
* **Reading Assigned**: W3C WAI-ARIA Overview - https://www.w3.org/WAI/standards-guidelines/aria/

**Week 2: HTML Foundations**

**August 26 (Tue) - Session 3**

* **PRE-CLASS VIDEOS**: "File Management via GitHub" (15 min) + "VS Code" (8 min)
* HTML structure and syntax
* Essential HTML elements: headings, paragraphs, lists
* Semantic HTML and why it matters
* **Lab**: Build a basic personal webpage

**August 28 (Thu) - Session 4**

* **PRE-CLASS VIDEO**: "HTML basics" (foundational concepts) (12 min)
* Links, images, and media elements
* HTML forms basics
* Introduction to accessibility (alt text, semantic structure)
* **Lab**: Enhance personal webpage with links and images
* **Discussion**: ARIA reading discussion - how does accessibility framework connect to the websites you analyzed?

**Week 3: More HTML & Intro to CSS**

**September 2 (Tue) - Session 5**

* **PRE-CLASS VIDEOS**: "CSS with/without" (3 min) + "Intro to CSS" (10 min)
* HTML tables and when to use them
* HTML validation and debugging
* Introduction to CSS: syntax and selectors
* **Lab**: Add basic CSS styling to HTML page

**September 4 (Thu) - Session 6**

* CSS properties: colors, fonts, spacing
* The box model
* CSS debugging with browser dev tools
* **Lab**: Style the personal webpage with CSS

**Week 4: CSS Layout & Design**

**September 9 (Tue) - Session 7**

* **PRE-CLASS VIDEO**: "UF Design Lecture" (responsive design concepts) (12 min)
* CSS layout: display properties, positioning
* Introduction to Flexbox
* **Lab**: Create flexible layouts
* **PROJECT 1 ASSIGNED**: Personal Portfolio Website (Due Week 6)

**September 11 (Thu) - Session 8**

* CSS Grid basics
* Combining Flexbox and Grid
* **Lab**: Practice with Grid layouts
* **Quiz 1**: HTML & Basic CSS
* **Reading Assigned**: "Web Accessibility for Developers" (Chapters 1-2) - https://open.umn.edu/opentextbooks/textbooks/678

**Week 5: Responsive Design & Advanced CSS**

**September 16 (Tue) - Session 9**

* Mobile-first responsive design principles
* Media queries and breakpoints
* Responsive images and media
* **Lab**: Make existing layouts responsive

**September 18 (Thu) - Session 10**

* CSS transitions and basic animations
* CSS custom properties (variables)
* **Lab**: Add animations and polish to designs
* **Threaded Discussion 2 Launch**: Mobile vs. desktop web experiences - analyze differences in 3 sites. How do accessibility considerations change across devices?

**Week 6: Project 1 Work & GitHub Pages**

**September 23 (Tue) - Session 11**

* Introduction to GitHub Pages deployment
* Git basics: add, commit, push
* **Lab**: Deploy first project to GitHub Pages
* **Discussion**: Web accessibility reading discussion - how do design decisions impact user experience?

**September 25 (Thu) - Session 12**

* Project 1 work time and troubleshooting
* Peer review and feedback
* **PROJECT 1 DUE**: Personal Portfolio Website (deployed on GitHub Pages)

**Week 7: Introduction to JavaScript**

**September 30 (Tue) - Session 13**

* What is JavaScript and why do we need it?
* JavaScript syntax: variables, data types, operators
* Console and debugging basics
* **Lab**: First JavaScript programs

**October 2 (Thu) - Session 14**

* Functions and scope
* Conditional statements and loops
* **Lab**: Interactive JavaScript exercises
* **Reading Assigned**: Benjamin, Ruha. "Introduction: The New Jim Code" from *Race After Technology* (2019)

**Week 8: JavaScript and the DOM**

**October 7 (Tue) - Session 15**

* What is the DOM?
* Selecting and manipulating HTML elements
* Event handling basics
* **Lab**: Add interactivity to web pages

**Fall Break - October 9 (Thu) - No Class**

**Week 9: Interactive Web Pages**

**October 14 (Tue) - Session 16**

* More DOM manipulation: creating and removing elements
* Form handling with JavaScript
* **Lab**: Build interactive forms
* **PROJECT 2 ASSIGNED**: Interactive Web Application (Due Week 12)

**October 16 (Thu) - Session 17**

* JavaScript objects and arrays
* Local storage basics
* **Lab**: Storing user data
* **Quiz 2**: CSS Layout & Basic JavaScript

**Week 10: Accessibility & User Experience**

**October 21 (Tue) - Session 18**

* Web accessibility principles (WCAG basics)
* ARIA labels and semantic HTML
* Testing for accessibility
* **Lab**: Audit and improve accessibility of existing projects

**October 23 (Thu) - Session 19**

* User experience fundamentals
* Form validation and error handling
* **Lab**: Improve form UX and validation
* **Threaded Discussion 3 Launch**: Accessibility on the web - analyze 3 sites using accessibility tools or screen readers. Connect to Benjamin's "New Jim Code" - how might automated systems create barriers?

**Week 11: APIs and External Data**

**October 28 (Tue) - Session 20**

* What are APIs?
* Introduction to fetch() and promises
* Working with JSON data
* **Lab**: Fetch data from a simple API
* **Discussion**: How does Benjamin's concept of "New Jim Code" apply to the data and algorithms we encounter in web APIs?

**October 30 (Thu) - Session 21**

* Google Sheets as a simple database
* Setting up Google Sheets API access
* **Lab**: Connect web page to Google Sheets data

**Week 12: Project 2 Work & Advanced JavaScript**

**November 4 (Tue) - Session 22**

* Error handling in JavaScript
* Async/await basics
* **Lab**: Refine API integrations

**November 6 (Thu) - Session 23**

* Project 2 work time and troubleshooting
* Code review and optimization
* **PROJECT 2 DUE**: Interactive Web Application (deployed on GitHub Pages)

**Week 13: Advanced Topics & Project 3 Prep**

**November 11 (Tue) - Session 24**

* CSS frameworks overview (Bootstrap, Tailwind concepts)
* Component-based thinking
* **Lab**: Create reusable CSS components
* **PROJECT 3 ASSIGNED**: Final Project - Comprehensive Web Application (Due Week 15)
* **Reading Assigned**: "Digital African Literatures and the Coloniality of Data" - https://www.cambridge.org/core/journals/cambridge-journal-of-postcolonial-literary-inquiry/article/abs/digital-african-literatures-and-the-coloniality-of-data/2201886F1ACCE556CE093E89B3F716BF

**November 13 (Thu) - Session 25**

* Performance basics: optimizing images, CSS, JavaScript
* Web standards and best practices
* **Lab**: Optimize previous projects

**Week 14: Final Project Work**

**November 18 (Tue) - Session 26**

* Final project planning and wireframing
* Advanced Git: branches, collaboration
* **Lab**: Project planning and initial setup

**November 20 (Thu) - Session 27**

* Final project work time
* Individual consultations
* **Quiz 3**: JavaScript, APIs, and Integration
* **Discussion**: How does the "coloniality of data" reading connect to our work with APIs and data integration? What power structures are embedded in the tools we use?

**Thanksgiving Break - November 25-27 - No Classes**

**Week 15: Final Project Completion**

**December 2 (Tue) - Session 28**

* Final project presentations
* Course reflection and next steps
* **PROJECT 3 DUE**: Final Project (deployed on GitHub Pages)
* **Threaded Discussion 4**: Reflect on your learning journey - what surprised you most about web development? How has your understanding of technology's role in society evolved through the readings and projects?

**Project Resources & Cultural Readings**

**Project 1: Personal Portfolio Website**

**Technical Resources:**

* MDN Web Development Learning Path - https://developer.mozilla.org/en-US/docs/Learn\_web\_development
* W3Schools HTML Tutorial - https://www.w3schools.com/html/
* GitHub Pages Quickstart Guide - https://docs.github.com/en/pages/quickstart
* GitHub Pages Deployment Tutorial - https://everhour.com/blog/how-to-host-website-on-github/

**Project 2: Interactive Web Application**

**Technical Resources:**

* MDN JavaScript Learning Path - https://developer.mozilla.org/en-US/docs/Learn\_web\_development/Core/Scripting
* Codecademy Interactive JavaScript Course - https://www.codecademy.com/catalog/language/html-css
* W3C ARIA Authoring Practices Guide - https://www.w3.org/WAI/ARIA/apg/
* W3C Web Accessibility Tutorials - https://www.w3.org/WAI/tutorials/

**Project 3: Data-Driven Web Application**

**Technical Resources:**

* Google Sheets API JavaScript Quickstart - https://developers.google.com/sheets/api/quickstart/js
* "Reading Public Google Sheets Data with JavaScript" Tutorial - https://medium.com/@ravipatel.it/step-by-step-guide-reading-public-google-sheets-data-using-javascript-and-displaying-it-on-an-html-f6aee8416a9c
* Google Sheets API Tutorial for Beginners - https://stateful.com/blog/google-sheets-api-tutorial
* Google Sheets API Overview & Documentation - https://developers.google.com/workspace/sheets/api/guides/concepts

**Unit Theoretical Readings**

**Unit 1 (Weeks 1-3): HTML & Web Foundations**

**Reading:** W3C WAI-ARIA Overview - https://www.w3.org/WAI/standards-guidelines/aria/  
*Introduces the framework for making web content accessible to all users*

**Unit 2 (Weeks 4-8): CSS Design & Responsive Layout**

**Reading:** "Web Accessibility for Developers" (Chapters 1-2) - https://open.umn.edu/opentextbooks/textbooks/678  
*Open access textbook connecting design decisions to user accessibility*

**Unit 3 (Weeks 9-12): JavaScript Interactivity**

**Reading:** Benjamin, Ruha. "Introduction: The New Jim Code" from *Race After Technology* (2019). Available through academic databases.  
*Examines how digital technologies can automate and amplify social inequalities*

**Unit 4 (Weeks 13-15): APIs & Data Integration**

**Reading:** "Digital African Literatures and the Coloniality of Data" - https://www.cambridge.org/core/journals/cambridge-journal-of-postcolonial-literary-inquiry/article/abs/digital-african-literatures-and-the-coloniality-of-data/2201886F1ACCE556CE093E89B3F716BF  
*Critical examination of how data systems and digital technologies intersect with power structures*

**Assessment Summary**

* **3 Projects**: 60% of grade
  + Project 1: Personal Portfolio (15%)
  + Project 2: Interactive Web Application (20%)
  + Project 3: Final Project (25%)
* **3 Quizzes**: 20% of grade
* **In-Class Labs & Participation**: 15% of grade
* **Threaded Discussions**: 5% of grade

**Key Learning Outcomes**

By the end of this course, students will be able to:

1. Create semantic, accessible HTML documents
2. Design responsive layouts using modern CSS
3. Add interactivity with JavaScript and DOM manipulation
4. Integrate external data sources using APIs
5. Deploy web applications using GitHub Pages
6. Apply web accessibility principles
7. Follow web development best practices and workflows
8. Critically analyze the social and cultural implications of web technologies

**Required Tools**

* Computer with internet access
* VS Code (free)
* GitHub account (free)
* Modern web browser (Chrome, Firefox, Safari, or Edge)

**Component Library**

Throughout all projects, students will develop and maintain:

* **CSS Reset/Base Styles**: Consistent starting point
* **Typography System**: Headings, body text, font choices
* **Color Palette**: Primary, secondary, accent colors
* **Button Styles**: Various button states and types
* **Form Components**: Input fields, labels, validation styles
* **Navigation Components**: Header/menu systems
* **Layout Utilities**: Grid systems, spacing utilities

This shared component approach helps students understand design systems while allowing creative freedom in individual projects.