

## Teaching Guidelines for Web Programming Technologies PG-DAC September 2021

---

**Duration: 108 hours** (48 classroom hours + 48 lab hours + 12 revision/practice hours)

**Objective:** To introduce the students to HTML, CSS, JavaScript, XML, JSON, Ajax, Node.js, Express.js, React, React-Redux, and practical relevance of all these technologies.

**Evaluation:** 100 marks

**Weightage:** Theory Exam – 40%, Lab exam – 40%, Internals – 20%

### Text Books:

- Fundamentals of Web Development, 1e, by Randy Connolly, Ricardo Hoar / Pearson
- MERN Quick Start Guide – Build web applications with MongoDB, Express.js, React, and Node by Eddy Wilson Iriarte Koroliova / Packt

### References:

- Internet & World Wide Web : How to Program by Paul Deitel, Henry Deitel & Abbey Deitel / Pearson Education
  - XML - How to Program by Deitel et al / Pearson Education
  - Ajax in Action by Dave Crane, Eric Pascarello / Dreamtech Press
  - JavaScript: The Good Parts by Douglas Crockford / O'Reilly
  - Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node by Vasan Subramanian / Apress
  - Web Application Security: A Beginner's Guide by Bryan Sullivan & Vincent Liu / Tata McGraw Hill
  - W3Schools Tutorials [ <https://www.w3schools.com/> ]
  - Mozilla Developer Network Web Development Tutorials [ [https://developer.mozilla.org/en-US/docs/Learn/Getting\\_started\\_with\\_the\\_web](https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web) ]
  - Curated Tutorial Links on ES6, React, etc. [ <https://github.com/markarikson/react-redux-links> ]
- 

(Note: Each Session is of 2 hours)

### Session 1: Architecture of Web

#### Lecture:

- Brief history of the Internet
- How does the Internet work?
- Internet Protocol; HTTP
- Domain Names; Domain Name Service servers
- HTTP Protocols
  - Difference between HTTP 1.0, HTTP 1.1, and HTTP 2.0
  - Methods – GET, POST, HEAD, PUT, DELETE, etc.
  - Status codes
  - Stateless nature of the protocol and HTTP Session
  - HTTPS
- Architecture of the Web
- Web servers – IIS, Apache server

**Lab:**

- Exploring different browsers
  - Mozilla Firefox, Google Chrome, Safari
- Exploring different text editors
  - Windows: Notepad++, Linux: Gedit or Vim or Emacs

**Session 2: HTML****Lecture:**

- Introduction to HTML5
- Introduction to basic HTML Tags
  - Alignment, Headings, Anchor, Paragraph, Image, Lists, Tables, and iFrames
- HTML5
  - New features in HTML5
  - New elements, new attributes, link relations, microdata, ARIA accessibility, objects, events, and Canvas tags
  - HTML5 Validation
  - Audio & Video Support
  - Geo-location Support
- HTML Forms & Controls
  - Input, Text Area, Radio Button, Checkbox, Dropdown, Submit, Reset, Button, etc.
- Introduction to Document Object Model (DOM)

**Lab:**

- Create a HTML form for building your resume.

**Session 3: Cascading Style Sheets (CSS)****Lecture:**

- Introduction to CSS, Styling HTML with CSS, Structuring pages with CSS,
- Inline CSS, Internal CSS, External CSS, Multiple styles, CSS Fonts
- CSS Box Model
- id Attribute, class Attribute
- HTML Style Tags
- Linking a style to an HTML document

**Lab:**

- Apply inline, internal, and external CSS to change colors of certain text portions, bold, underline, and italics certain words in the previously created HTML resume form.

**Session 4: JavaScript****Lecture:**

- Introduction to JavaScript
- Variables in JavaScript
- Statements, Operators, Comments, Expressions, and Control Structures
- JavaScript Scopes
- Strings, String Methods
- Numbers, Number Methods
- Boolean Values
- Dates, Date Formats, Date Methods
- Arrays, Array Methods

**Lab:**

- Practice writing basic JavaScript programs for better understanding of the language constructs

## **Sessions 5 & 6: JavaScript**

### **Lecture:**

- Objects, Object Definitions, Object Properties, Object Methods, Object Prototypes
- Functions, Function Definitions, Function Parameters, Function Invocation, Function Closures
- Introduction to Object Oriented Programming in JS
  - Method, Constructor, Inheritance, Encapsulation, Abstraction, Polymorphism

### **Lab:**

- Write a JavaScript program to sort a list of elements by implementing a sorting algorithm.
- Write a JavaScript program to list the properties of a JavaScript object.

## **Sessions 7 & 8: JavaScript**

### **Lecture:**

- Document Object Model (DOM)
  - Object hierarchy in JavaScript
  - HTML DOM, DOM Elements, DOM Events
  - DOM Methods, DOM Manipulation
- Forms, Forms API, Forms Validation
- Regular Expressions
- Errors, Debugging
- Introduction to Browser Dev Tool
- Pushing code quality via JSLint tool

### **Lab:**

- Write a JavaScript function to get First and Last name from the previously created Resume form
- Validate the entire Resume form using client-side JavaScript
- Write a JavaScript function to validate whether a given value is RegEx or not.

## **Session 9: jQuery**

### **Lecture:**

- Introducing to jQuery
- jQuery selectors
- jQuery events
- jQuery animation effects
- jQuery DOM traversal and manipulation
- Data attributes and templates
- jQuery DOM utility functions
- jQuery plugins

### **Lab:**

- Write a jQuery program to get a single element from a selection of elements of a HTML page.
- You are having sample data for the link. Write jQuery code to change the hyperlink and the text of an existing link.
- Write a jQuery program to attach a click and double-click events to all <p> elements.
- Write a jQuery program to hide all headings on a page when they are clicked.
  - Also find the position of the mouse pointer relative to the left and top edges of the document.

## **Sessions 10 & 11: JSON & Ajax**

### **Lecture:**

- JSON: JavaScript Object Notation (JSON)
  - Introduction and need of JSON
  - JSON Syntax Rules

- JSON Data - a Name and a Value,
- JSON Objects, JSON Arrays, JSON Files
- JSON parsing
- Ajax
  - Introduction to Ajax
  - Ajax Framework
  - Ajax Architecture
  - Web services and Ajax
  - Ajax using JSON and jQuery

**Lab:**

- Create a page showing live score/feed using Ajax and JSON from a live sport/news service end-point given by the faculty

**Session 12: Introduction to Node.js**

**Lecture:**

- Introduction to Node.js
- Browser JS vs. Node.js
- ECMAScript 2015 (ES6)
- Node.js REPL

**Lab:**

- Install Node.js 12.x.x LTS version on your machine
- Write a recursive function in Node.js
- Write a Node program that prints all the numbers between 1 and 100, each on a separate line.  
A few caveats:
  - if the number is divisible by 3, print "foo"
  - if the number is divisible by 5, print "bar"
  - if the number is divisible by both 3 and 5, print "foobar"

**Sessions 13 & 14: Node.js Asynchronous Programming**

**Lecture:**

- Introduction to Asynchronous programming and callbacks
- Promises and async & await
- The Event Loop and Timers

**Lab:**

- Assignment on JavaScript callback functions
- Assignment on Timers, Promises, and Async & Await

**Session 15: Node.js Modules**

**Lecture:**

- Understanding Node modules, exports, and require
- Introduction to npm
  - package.json and package-lock.json files
  - Install, update, and manage package dependencies
  - Local and global packages

**Lab:**

- Create a module and import it in other programs
- Install a module/package using npm

**Session 16: Node.js Modules – *fs* and *http***

**Lecture:**

- File I/O – Sync & Async Methods

- HTTP Module – Building an HTTP server
- Developing a Node web application

**Lab:**

- Write a program to create a new file and write some content to it in synchronous mode and read and display file contents on standard output in async mode
- Build a simple Node.js web application serving both HTTP GET and POST methods

**Session 17: Introduction to Express**

**Lecture:**

- Introduction to Express
- Getting started with Express
- Application, Request and Response Objects
- Routes and Middlewares
- Templates, Template Engines, and Rendering Views

**Lab:**

- Use Node and Express to write a simple web application that consists of at least 5 route implementations
- Rebuild any previous Node assignment using Express and a template engine

**Session 18: Introduction to React**

**Lecture:**

- Introduction to React
- Getting started with React
- React Elements and React Components
- Function and Class Components
- Working with React Components and Props
  - Compose components
  - Render components
  - Declutter components

**Lab:**

- Rebuild any previous plain HTML lab assignment using React
- Build a React Clock app showing time (hh:mm:ss) of any three countries

**Sessions 19 & 20: React**

**Lecture:**

- Introduction to State and Lifecycle
- Stateful components and lifecycle methods
- Props vs. State vs. Context
- Handling events
- Conditional rendering

**Lab:**

- Implement the following items in the React Clock app
  - Update the time (hh:mm:ss) using State and Lifecycle methods
  - Add a close function on each rendered clock component
  - Assign background color of rendered clock components based on AM, PM

**Session 21: React**

**Lecture:**

- Lists and Keys
  - Rendering Multiple Components

- Basic List Component
- Working with forms and inputs
- Refs and the DOM
- Lifting state up

**Lab:**

- Implement and integrate a new feature in the React Clock app where one can select a country time zone from dropdown list and click on “Add” button to render it.

**Session 22: React**

**Lecture:**

- Error Boundaries
- Composition vs. Inheritance
  - Containment
  - Specialization
- Thinking in React

**Lab:**

- Implement error boundaries at appropriate places in the React Clock app

**Session 23: Introduction to React-Redux**

**Lecture:**

- Introduction to Redux
- Actions, Reducers, and Stores
- Usage with React

**Lab:**

- Make necessary changes in the design and implementation of React Clock app using React-Redux to maintain the application state.

**Session 24: Responsive Web Design & Web Security**

**Lecture:**

- Introduction of UI Scripting
- The Best Experience for All Users
  - Desktop, Tablet, Mobile
- Bootstrap
  - Overview of Bootstrap, Need to use Bootstrap
  - Bootstrap Grid System, Grid Classes, Basic Structure of a Bootstrap Grid
  - Typography
  - Components – Tables, Images, Jumbotron, Wells, Alerts, Buttons, Button Groups, Badges/Labels, Progress Bars, Pagination, List Groups, Panels, Dropdowns, Collapse, Tabs/Pills, Navbar
  - Forms, Inputs
  - Bootstrap Themes, Templates
- Web Security
  - Introduction to Web security
  - SQL Injection, Cross-Site Scripting (XSS)
  - JSON and Security Concerns, Cross Site Request Forgery (CSRF), Injection Attacks
  - Security Standards (OWASP)

**Lab:**

- Update the design of the Resume form using Bootstrap