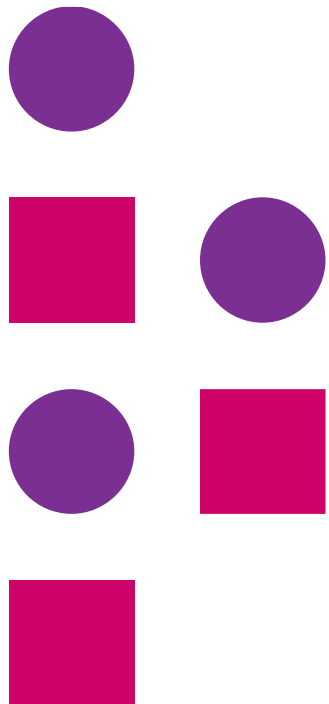


FULL STACK WEB DEVELOPER COURSE SYLLABUS

WWW.WEBSTACKACADEMY.COM



COURSE OBJECTIVES

- Enable participants to develop a complete web application from the scratch that includes Front-end, Back-end and Data-exchange technologies
- Build strong foundations (ex: OOPS) in entry level engineers / working professionals thereby making them job ready as per latest industry requirements. Enable them to learn new technologies by applying foundation paradigms
- By the end of the program participants will be become an industry-ready engineer who can be readily deployed in a project

COURSE FLOW

WSA Full Stack Web developer course starts with introducing participants with front-end technologies (ex: HTML5) and makes them develop responsive one page web application. Followed by that it deep dives into MERN stack components and brings hands-on perspective to building scalable web applications by leveraging “JavaScript everywhere” phenomenon. Throughout the course foundation paradigms are stressed to ensure participants clearly understand software engineering fundamentals and apply them.

The course further covers various data-exchange mechanisms (ex: HTTP) to get complete perspective of Full Stack. DISHA sessions are interleaved throughout the program for resume and interview preparation aspects for placements.

MODULE TOPICS

- Front-end (HTML5 | CSS3 | Bootstrap | JavaScript | jQuery)
- Foundation paradigms (OOPS | Design Patterns | Object Modelling | JSON | AJAX)
- MEAN Stack (MongoDB | Express Framework | ReactJS | Node.js)
- Data Exchange (HTTP | JSON | REST APIs)
- Development Environment & Tools
- DISHA (Resume & Interview prep package)

DURATION

6 months

PLATFORM

- Linux (ex: Ubuntu) or Windows Host system
- Firefox, Chrome Browsers

PREREQUISITES

B.E / B.Tech / M.E / M.Tech / MCA

DELIVERY METHOD

Live Virtual Classes and Intervention Sessions
(Assignment / Project Explainer, Live Doubt Clearing and Evaluation)

Front-end Course Modules

Hyper Text Mark-up Language (HTML5)

Module Objectives:

- Building Strong expertise to develop front end application using HTML5
- Implement MVC and responsive design to scale well across PC, tablet and Mobile Phone

Overview:

This course provides you hands-on experience and exposure to developing HTML5 based single page application for browsers. This course builds strong foundation on HTML5 which will help developer to use HTML5 concepts for building responsive web application.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed Course Contents:

- Introduction HTML
- HTML Basics
- HTML Elements
- HTML5 Semantic
- HTML Attributes
- HTML Headings
- HTML Paragraph
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Computer Code
- HTML Comments & Colours
- HTML CSS, Links and Images
- HTML Lists
- HTML Blocks
- HTML Classes
- HTML Layout
- HTML Responsive
- HTML iframes
- HTML JavaScript
- HTML Head
- HTML Entities and URI Code
- HTML Symbols and XHTML
- HTML Charset and Forms

Cascading Style Sheets (CSS3)

Module Objectives:

- Building Strong expertise to develop front end application using CSS3
- Implement MVC and responsive design to scale well across PC, tablet and Mobile Phone

Overview:

This course provides you hands-on experience and exposure to developing CSS3 based web application. This course builds strong foundation on CS33 which will help developer to use CSS3 concepts for building responsive web application.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

- Introduction CSS3
- CSS3 Syntax
- CSS3 How To
- CSS3 Colours
- CSS3 Backgrounds
- CSS3 Borders
- CSS Padding
- CSS Height/Width
- CSS3 Gradients
- CSS3 Shadows
- CSS3 Text
- CSS3 Fonts
- CSS3 2D Transforms
- CSS3 3D Transforms
- CSS Links
- CSS Lists
- CSS Tables
- CSS Box Model
- CSS Outline
- CSS Display
- CSS Max-width
- CSS Position
- CSS Float
- CSS Inline-block
- CSS Align
- CSS Combinators
- CSS Pseudo-class
- CSS Pseudo-element
- CSS Navigation Bar
- CSS Dropdowns
- CSS Tooltips
- CSS3 Images
- CSS Attr Selectors
- CSS Forms
- CSS Counters
- CSS3 Animations
- CSS3 Buttons
- CSS3 Pagination
- CSS3 Multiple Columns
- CSS3 User Interface
- CSS3 Box Sizing
- CSS3 Filters
- CSS3 Media Queries
- CSS3 Responsive

Bootstrap

Module Objectives:

- To become proficient in Bootstrap concepts
- To develop a web pages based on Bootstrap

Overview:

This course is targeted for fresh engineers or professional who want to build competency in the Bootstrap Based web development.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

- Introduction to Bootstrap
- Bootstrap Basics
- Bootstrap Grids
- Bootstrap Themes
- Bootstrap CSS
- Bootstrap JS

JavaScript

Module Objectives:

- Understand core features of JavaScript
- Build interactive and user-friendly frontend applications using HTML, CSS and JavaScript
- Apply OOP concepts by learning JavaScript
- Debug frontend applications using Google Chrome debugger

Overview:

This course provides you hands-on experience and exposure to develop frontend application using JavaScript. Starting with introduction, this course deep dives into core features like Event handling, Form handling and Regular expressions. Specific focus is given for Document Object Model (DOM) and manipulating HTML using various DOM APIs. Along with building JavaScript features this course builds sound foundations in Algorithms, Problem solving and Debugging techniques which is critical for a web application developer.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- VSCode editor with Live server
- Google Chrome Debugger

Prerequisites:

- HTML and CSS
- Basic understanding of OOPS

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:**Algorithms and Problem solving**

- Problem solving - what?
- Introduction to SDLC
- Polya's rules
- Algorithm design methods
- Pseudo code creation
- Flowcharts

Introduction to JavaScript

- History of JavaScript
- Advantages
- Limitations
- Script element
- Creating your first JavaScript program
- Coding convention
- Setting up development environment (with VSCode)

Types and Statements

- Keywords in JavaScript
- Overview of Data types
- Primitive Data types
- Non-primitive Data types
- Conditional statements
- I/O in JavaScript
- Loops

Operators

- Introduction to operators
- Operator precedence and associativity
- Deep dive into operators
- Arithmetic
- Comparison
- Ternary
- Logical
- Language
- Bitwise

Functions - Level I

- Introduction to functions
- Function definition
- Passing values
- Returning values
- Robust parameter handling
- Local and global variables
- Functions as objects
- Function constructor

Functions - Level II

- Function invocation patterns
- Recursion functions
- Generator functions
- Arrow functions
- Variadic functions
- JavaScript scopes
- Function closures

Arrays and Strings

- Introduction to Arrays
- Array declaration
- Array access methods
- Multi-dimensional arrays
- String properties
- String access methods

Regular Expressions

- Introduction to RegExp
- Regular expression usage
- Modifiers
- RegExp patterns
- RegExp methods
- String methods for RegExp
- Type conversion in JavaScript

Objects in JavaScript

- Introduction to objects
- Type of objects in JavaScript
- Creating objects
- Object methods
- Constructor function
- Prototype in JavaScript
- Inheritance using prototype chain

Event handling

- JavaScript events
- Event handler
- Event flow
- Event bubbling and capturing
- Event listeners
- Event types

Document Object Model (DOM)

- Introduction to DOM
- Types of DOM
- DOM standards and methods
- Manipulating documents using DOM
- Handling images
- Table manipulation
- Animation
- Node and Node-list handling

Browser Object Model (BOM)

- Introduction to BOM
- DOM vs BOM differences
- Window object and methods
- BOM navigator
- BOM history
- BOM location
- BOM timer
- Introduction to Cookies
- Session and persistent cookies

Form Handling

- Introduction to forms
- Form processing
- Forms object
- Accessing data from forms
- Form validation
- Additional features in forms
- Validation APIs

Debugging Techniques

- JavaScript Errors
- Error handling mechanisms
- Introduction to Google Chrome debugger
- Deep dive into debugger window
- Introduction to Breakpoints
- Changing variable values in runtime
- Avoiding mistakes

jQuery**Module Objectives:**

- Building Strong expertise to develop front end application using HTML5, CSS3 and JavaScript along with jQuery and Angular framework
- Implement MVC and responsive design to scale well across PC, tablet and Mobile Phone

Overview:

This course provides you hands-on experience and exposure to develop jQuery Mobile based responsive web application. It builds strong foundation of jQuery which will help developer to apply concepts for responsive web frontend development.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

- Introduction to jQuery

- jQuery Syntax
- jQuery Selectors
- jQuery Events
- jQuery Effects
- jQuery HTML
- jQuery Traversing
- jQuery AJAX & Misc

MERN Stack – Course modules

MongoDB

Module Objectives:

- Building strong expertise in Document Oriented Non-Relational Database Management System
- Implement frontend and backend scenarios to read, write and update data stored in MongoDB

Overview:

This course provides you hands-on experience and exposure to implement frontend and backend scenarios to read, write and update data using MongoDB. This course builds strong foundation for web application development based on client-server architecture.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

CRUD Operations Using MongoDB

- Installing MongoDB
- Connecting to MongoDB
- Schemas
- Models
- Saving a Document
- Querying Documents
- Comparison Query Operators
- Logical Query Operators
- Regular Expressions
- Counting
- Pagination
- Updating Documents- Query First
- Updating a Document- Update First
- Removing Documents

Mongoose - Data Validation

- Validation
- Built-In Validators
- Custom Validators
- Async Validators
- Validation Errors
- SchemaType Options

Mongoose- Modelling Relationships Between Connected Data

- Referencing Documents
- Population
- Embedding Documents
- Using an Array of Sub-documents
- Transactions
- Object ID
- Validating Object ID's

Authentication and Authorization

- Introduction
- Creating the User Model
- Registering Users
- Using Lodash
- Hashing Passwords
- Authenticating Users
- Testing the Authentication
- JSON Web Tokens
- Generating Authentication Tokens
- Storing Secrets in Environment Variables
- Setting Response Headers
- Encapsulating Logic in Mongoose Models
- Authorization Middleware
- Protecting Routes
- Getting the Current User
- Logging Out Users
- Role Based Authorization
- Testing the Authorization

Handling and Logging Errors

Deployment

- Introduction
- Preparing the App for Production
- Getting Started With Heroku
- Preparing the App for Deployment
- Adding the Code to a Git Repository
- Deploying to Heroku
- Viewing Logs
- Setting Environment Variables
- MongoDB in the Cloud

Express Framework

Module Objectives:

- Building Strong expertise on Express framework to develop responsive web application
- Implement MVC and responsive design to provide unified and intuitive user experience across PC, tablet and Mobile Phone

Overview:

This course provides you strong foundation on Express framework driven by hands-on experience and exposure to develop responsive web application using MVC design pattern. During the course, you should be able to builds strong background about Express and Node.js architecture, usage of services and applying Node.js concept to develop frontend application supported by backend web servers.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites:

Academic level web application knowledge

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

Express Framework - (Building RESTful API's)

- Introduction
- RESTful Services
- Introducing Express
- Building Your First Web Server
- Nodemon
- Environment Variables
- Route Parameters
- Handling HTTP GET Requests
- Handling HTTP POST Requests
- Calling Endpoints Using Postman
- Input Validation
- Handling HTTP PUT Requests
- Handling HTTP Delete Requests

Express - Advanced Topics

- Middleware
- Creating Custom Middleware
- Built-In Middleware
- Third-party Middleware
- Environments
- Configuration
- Debugging

Asynchronous JavaScript

- Synchronous Vs. Asynchronous Code
- Patterns For Dealing With Asynchronous Code
- Callbacks
- Callback Hell
- Named Functions to Rescue
- Promises
- Replacing Callbacks with Promises
- Consuming Promises
- Creating Settled Promises
- Running Parallel Promises
- Async and Await

ReactJS

Module Objectives:

- Create Single Page Applications (SPA) using React
- Learn core React features and use-cases
- Integrate React Applications into Firebase backend
- Deploy React applications into popular servers

Overview:

This course provides you building SPA using React and its core features by taking a hands-on approach. Starting with Introduction this course deep dives into core React features like components, state and event binding and rendering. Followed by core features, participants will learn to integrate React with backend services (ex: Firebase) and consuming HTTP services in forms of REST APIs. The course exposes participants with advanced features like Routing, Authentication and authorization followed by deploying into popular servers (ex: GITHUB)

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome Browsers
- Firebase (Backend integration)

Prerequisites:

- OOPS concepts
- Knowledge of HTML, CSS and JavaScript

Delivery method:

Instructor lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

Introduction to React

- History of React
- Key Benefits of React
- React development environment
- Creating your first React Application
- React Source code structure

JSX

- Introduction to JSX
- Coding in JSX
- Expressions in JSX

- Working with HTML
- Conditional Constructs

Components

- Introduction to components
- Why Components?
- Writing JSX code in components
- Adding CSS
- Populating Data Dynamically
- Passing data through "props"
- Multiple Components

State and Event Binding

- Introduction to Events
- Event Handlers
- Working with state
- Data Binding
- Controlled and Uncontrolled Components

Rendering Lists and Conditional Contents

- Rendering lists of data
- Using stateful list
- Keys in data
- Conditional Contents
- Adding dynamic styles

Debugging Techniques

- Understanding error message
- Code flow and warnings
- Breakpoints
- Using React Dev Tools

Class based Components

- What and Why?
- Adding first class based component
- Working with states and events
- Component Life cycle and in action
- Class based vs functional component

HTTP

- Introduction to HTTP
- Methods in HTTP
- Response code in HTTP
- Introduction to REST interfaces
- Characteristics of REST
- Introduction to JSON
- JSON data representation
- GET request
- Using async and await
- Handling Http errors
- useEffects()
- POST request

Custom React Hooks

- What are custom hooks?
- Creating and using custom hooks
- Custom HTTP hooks

Forms

- Introduction to Forms
- Working with user input and submission
- Adding validation
- Working with custom hooks

Routing and Navigation

- Introduction to Single Page Applications
- What is Routing?
- Why Routing?
- Installing react router
- Defining and using routes
- Working with links
- Dynamic routes with Params
- Nested routes
- Redirecting the user
- Working with query Param
- Sending and getting data via HTTP

Authentication

- What is authentication?
- Why and How?
- Authentication tokens
- Setting up
- Adding signup
- Showing feedback to the user
- Adding User login
- Managing authstate with context
- Using tokens
- Redirecting the user
- Adding logout
- Protecting frontend pages
- Persisting User authentication status
- Adding auto logout

Deployment

- Introduction
- Deployment steps
- Adding lazy loading
- Building the code for production
- Getting started with deployment
- Handling routes and finishing deployment

Node.js

Module Objectives:

- Building Strong expertise on Node.js core to develop web application
- Implement web application and deployment

Overview:

This course provides you strong foundation on Node.js driven by hands-on experience and exposure to develop NodeJS based application and deployment. During the course you should be able to builds strong background about Node.js architecture, usage of services and applying Node.js concept to develop frontend application supported by backend web servers.

Platform:

- Linux (Fedora / Mandriva / Ubuntu) or Windows Host system
- Firefox, Chrome

Prerequisites

Academic level web application knowledge

Delivery method:

Instructed lead, hands-on exercises backed with assignments and mini project

Detailed course contents:

Overview and Architecture

- What is Node
- Node Architecture
- How Node Works
- Installing Node
- Your First Node Program
- Event Loop

Node Module System

- Introduction
- Global Object
- Modules
- Creating a Module
- Loading a Module
- Module Wrapper Function
- Path Module
- OS Module
- File System Module
- Events Module
- Event Arguments
- Extending Event Emitter
- HTTP Module

Node Package Manager (NPM)

- Introduction
- Package.json
- Installing a Node Package
- Using a Package

- Package Dependencies
- NPM Packages and Source Control
- Semantic Versioning
- Listing the Installed Packages
- Viewing Registry Info for a Package
- Installing a Specific Version of a Package
- Updating Local Packages
- DevDependencies
- Uninstalling a Package
- Working with Global Packages
- Publishing a Package
- Updating a Published Package



WEB STACK ACADEMY

#83, Farah Towers,
1st floor, MG Road,
Bangalore - 560001

+91-809 555 7 333
+91-80-4128 9576
training@webstackacademy.com
www.webstackacademy.com

