Bright Mentors®

Java Full Stack Development

Training Program

Duration: 6 Months

www.brightmentors.in

Core Java:

Module 1: Introduction To Java

- ✓ Evolution and History of Java
- ✓ Key Features of Java
- ✓ Java Versions and Updates
- ✓ Java Platforms and Their Applications
- ✓ Understanding JDK, JRE, and JVM
- ✓ JVM Architecture and Its Components
- ✓ Real-World Applications of Java
- ✓ Installing Java on Windows and macOS
- ✓ Configuring System Path for Java

Module 2: Writing Your First Java Program

- ✓ Overview of Text Editors and IDEs
- ✓ Understanding Java Keywords and Identifiers
- ✓ Naming Rules and Conventions in Java
- ✓ Using Comments for Code Clarity
- ✓ Writing and Structuring Java Source Code
- ✓ Compiling Java Programs and Generating Bytecode
- ✓ Executing Java Programs and Understanding Output

Module 3: WORKING WITH METHODS IN JAVA

- ✓ Introduction to Methods and Their Purpose
- ✓ Method Signature and Structure
- ✓ Implementing Methods with Code Blocks
- ✓ Defining and Using Parameterized Methods
- ✓ Invoking Methods in Java
- ✓ Understanding the Main Method and Its Role

Module 4: Variables And Data Types

- ✓ Understanding Variables in Java
- ✓ Declaring and Initializing Variables
- ✓ Data Types in Java: Primitive and Non-Primitive
- ✓ Assigning Values to Variables
- ✓ Type Casting in Java
- ✓ Implicit vs. Explicit Type Casting

Module 5: Operators In Java

- ✓ Overview of Java Operators
- ✓ Unary, Binary, and Ternary Operators
- ✓ Bitwise Operators and Their Applications
- ✓ Understanding Operator Return Types

Module 6: Control Statements In Java

- ✓ Conditional Statements: If-Else and Switch Case
- ✓ Iteration Statements: Loops in Java
- ✓ Understanding Infinite Loops
- ✓ Break and Continue Statements
- ✓ Nested Loops and Their Usage
- ✓ Practical Loop Exercises for Hands-On Learning

Module 7: Object-oriented Programming (Oop) In Java

- ✓ Introduction to Object-Oriented Programming (OOP)
- ✓ Understanding Classes and Objects
- ✓ Core Pillars of OOP
    - Encapsulation
    - Inheritance
    - Polymorphism
    - Abstraction

Module 8: Encapsulation In Java

- ✓ Understanding Encapsulation and Its Importance
- ✓ Defining Classes and Instance Variables
- ✓ Implementing Constructors in Java
- ✓ Using Getters and Setters for Data Access
- ✓ Instantiating Classes and Creating Objects
- ✓ Understanding Object References and Reference Variables
- ✓ Utilizing the this Keyword in Java
- ✓ Working with Static Members:
  - Static Blocks
  - Static Variables
  - Static Methods

Module 9: Constructors In Java

- ✓ Introduction to Constructors and Their Role
- ✓ Types of Constructors:
  - Default Constructor
  - Parameterized Constructor
  - Copy Constructor
- ✓ Constructor Overloading and Its Benefits
- ✓ Understanding Initialization Blocks
- ✓ Implementing Constructor Chaining in Java

Module 10: Inheritance In Java

- ✓ Understanding Inheritance and Its Significance
- ✓ Real-World Use Cases of Inheritance
- ✓ Types of Inheritance in Java
- ✓ Access Modifiers and Their Role in Inheritance
- ✓ Utilizing the super Keyword and super Method
- ✓ Understanding the final Keyword in Java
- ✓ Method Overriding and Its Importance
- ✓ Rules and Best Practices for Method Overriding

Module 11: Polymorphism In Java

- ✓ Introduction to Polymorphism and Its Advantages
- ✓ Types of Polymorphism in Java
- ✓ Implementing Method Overloading
- ✓ Rules and Best Practices for Method Overloading

Module 12: Abstraction In Java

- ✓ Understanding Abstraction and Its Role in OOP
- ✓ Defining and Implementing Abstract Methods
- ✓ Working with Abstract Classes
- ✓ Introduction to Interfaces in Java
- ✓ Implementing Interfaces in Java Applications
- ✓ Comparing Abstract Classes and Interfaces

Module 13: Nested Classes And Interfaces

- ✓ Introduction to Nested (Inner) Classes
- ✓ Implementing Static Nested Classes
- ✓ Understanding Non-Static Nested Classes
- ✓ Local Nested Classes and Their Use Cases
- ✓ Defining and Using Anonymous Nested Classes

Module 14: AGGREGATION AND COMPOSITION IN JAVA

- ✓ Understanding Association in Java
- ✓ Types of Association
- ✓ Implementing Aggregation in Java
- ✓ Understanding Composition and Its Applications

Module 15: Packages And Jar Files In Java

- ✓ Introduction to Packages in Java
- ✓ Creating and Managing Packages

- ✓ Commands to Create a JAR File
- ✓ Using the import Statement in Java
- ✓ Understanding Static Import in Java
- ✓ Associating Classes with Packages

Module 16: Arrays In Java

- ✓ Introduction to Arrays and Their Uses
- ✓ Declaring and Initializing Arrays
- ✓ Memory Allocation for Arrays
- ✓ Accessing and Manipulating Array Elements
- ✓ Different Types of Arrays in Java
- ✓ Understanding Array References
- ✓ Passing Array References to Methods
- ✓ Working with Variable-Length Arguments

Module 17: String Handling In Java

- ✓ Introduction to Strings in Java
- ✓ Understanding String Literals
- ✓ String Memory Management
- ✓ Exploring the String Class and Its Methods
- ✓ Instantiating the String Class
- ✓ Understanding the String Constant Pool
- ✓ Working with the StringBuffer and StringBuilder Classes

Module 18: Exception Handling In Java

- ✓ Understanding Exceptions and Their Importance
- ✓ Hierarchy of Java Exception Classes
- ✓ Overview of the Throwable Class
- ✓ Differentiating Error and Exception Classes
- ✓ Key Differences: Errors vs. Exceptions
- ✓ Creating and Throwing Custom Exceptions
- ✓ Using throw and throws Keywords in Java
- ✓ Handling exception with try, catch and finally blocks

Module 19: Multithreading In Java

- ✓ Introduction to Multitasking in Java
- ✓ Process-Based Multitasking (Multiprocessing)
- ✓ Thread-Based Multitasking (Multithreading)
- ✓ Understanding Threads and Their Role in Java Applications
- ✓ Java Thread Scheduler and Its Functionality
- ✓ Life Cycle of a Thread
- ✓ Approaches to Creating Multithreading in Java
- ✓ Using the Thread Class and Runnable Interface
- ✓ Understanding Daemon Threads
- ✓ Implementing Inter-Thread Communication
- ✓ Avoiding and Managing Deadlocks in Java
- ✓ Thread Synchronization Techniques:
  - Synchronized Methods
  - Synchronized Blocks
  - Synchronized Classes/Objects

Module 20: Garbage Collection In Java

- ✓ Introduction to Garbage Collection and Its Importance
- ✓ Understanding the Garbage Collector in Java
- ✓ Identifying Unreachable Objects
- ✓ Criteria for Garbage Collection Eligibility
- ✓ Techniques to Make an Object Eligible for Garbage Collection
- ✓ Methods to Request JVM to Execute Garbage Collection
- ✓ Using the gc() Method of the System Class
- ✓ Understanding the finalize() Method of the Object Class

Module 21: Key Features Of Java 8

- ✓ Introduction to Java 8 Enhancements
- ✓ Functional Interfaces and Their Usage
- ✓ Implementing Lambda Expressions
- ✓ Understanding Method References
- ✓ Exploring the Streams API for Efficient Data Processing

- ✓ Enhanced Date and Time API in Java 8
- ✓ Working with the Optional Class
- ✓ Miscellaneous Enhancements in Java 8

Advanced Java:

Module 1: Collection Framework In Java

Understanding Collections in Java

- ✓ Introduction to the Collection Framework
- ✓ Exploring the Generics Feature in Java
- ✓ Overview of Wrapper Classes in Java

Collection Interfaces and Types

- ✓ Understanding the Collection Interface
- ✓ Different Types of Collections in Java

List Collection and Its Implementations

- ✓ Overview of the List Collection
- ✓ ArrayList Class and Its Features
- ✓ LinkedList Class and Its Applications
- ✓ Vector Class and Its Usage
- ✓ Comparison: ArrayList vs. Vector
- ✓ Understanding the Stack Class

Set Collection and Its Implementations

- ✓ Overview of the Set Collection
- ✓ Working with the HashSet Class
- ✓ Internal Mechanism of HashSet
- ✓ Features of the LinkedHashSet Class
- ✓ Implementing TreeSet for Sorted Data

Queue Collection and Its Implementations

- ✓ Introduction to the Queue Collection
- ✓ Understanding the PriorityQueue Class
- ✓ Implementing ArrayDeque for Efficient Queue Operations

MAP INTERFACE AND ITS IMPLEMENTATIONS

- ✓ Introduction to the Map Interface in Java
- ✓ Understanding the HashMap Class
- ✓ Internal Working Mechanism of HashMap
- ✓ Overview of the Hashtable Class
- ✓ Key Differences: HashMap vs. Hashtable
- ✓ Exploring the LinkedHashMap Class for Ordered Mapping
- ✓ Implementing TreeMap for Sorted Key Storage
- ✓ Understanding the ConcurrentHashMap for Thread-Safe Operations

COMPARISON AND STREAM API USAGE IN COLLECTIONS

- ✓ Comparable vs. Comparator Interface: Key Differences and Use Cases
- ✓ Leveraging the Stream API for Efficient Collection Processing

Module 2: Jdbc And Sql

- ✓ Introduction to SQL (Structured Query Language)
- ✓ Understanding Databases and Their Role in Applications
- ✓ Overview of SQL Commands: DDL, DML, and DQL
- ✓ Introduction to JDBC (Java Database Connectivity)
- ✓ Importance and Need for JDBC in Java Applications
- ✓ Exploring the JDBC API Interface
- ✓ Understanding the Driver Interface and Driver Class
- ✓ Internal Working of the Driver Class
- ✓ Establishing a Database Connection in Java
- ✓ Creating and Using a Statement Object
- ✓ Working with the ResultSet Interface to Retrieve Data

Module 3: Servlets And Jsp – Web Development With Java

- ✓ Understanding Web Applications and Their Functionality
- ✓ Developing Web Applications Using Servlet Technology
- ✓ Exploring Advanced Java Concepts for Web Development
- ✓ Mastering Java with Data Structures and Algorithms (DSA)

Servlets in Java

- ✓ Introduction to Servlets and Their Role in Web Development
- ✓ Creating and Implementing Servlets in Java
- ✓ Understanding the Servlet Life Cycle
- ✓ Handling Client Requests and Server Responses

Java Server Pages (JSP)

- ✓ Building Web Applications with JSP Technology
- ✓ Overview of JSP (Java Server Pages) and Its Benefits
- ✓ JSP Life Cycle and Execution Flow
- ✓ Utilizing Implicit Objects in JSP for Efficient Development
- ✓ Understanding and Implementing JSP Tags
- ✓ Working with JSTL (JSP Standard Tag Library)

Module 4: Design Patterns – Structured Software Development In Java

- ✓ Understanding Design Patterns and Their Role in Software Architecture
- ✓ Exploring Different Categories of Design Patterns in Java

Types of Design Patterns in Java

- ✓ **Creational Design Patterns** – Object Creation Mechanisms and Best Practices
- ✓ **Structural Design Patterns** – Organizing Classes and Objects for Scalability
- ✓ **Behavioral Design Patterns** – Managing Object Interactions and Communication
- ✓ **Best Practices:** When and How to Implement Design Patterns for Optimized Development

Java Framework:

Module 1: Hibernate Framework – Object-relational Mapping In Java

- ✓ Introduction to Hibernate and Its Importance
- ✓ Overview of JPA (Java Persistence API) and Its Role in Persistence
- ✓ Understanding ORM (Object-Relational Mapping) Tools
- ✓ Introduction to Build Tools for Java Development
- ✓ Working with Maven for Dependency Management
- ✓ Creating a Maven Project and Configuring the POM.xml File
- ✓ Managing Project Dependencies in Maven

Hibernate Architecture and Core Components

- ✓ Understanding Hibernate Architecture and Its Components
- ✓ Key Building Blocks of Hibernate
  - SessionFactory, Session, Transaction, and Query Interfaces
- ✓ Creating a Persistent Class in Hibernate
- ✓ Configuring Hibernate with HBM and CFG Files
- ✓ Using Mapping Annotations for Entity Configuration
- ✓ Implementing Association Mapping in Hibernate
- ✓ Exploring Inheritance Strategies in Hibernate
- ✓ Understanding Hibernate Caching Mechanisms for Performance Optimization

Module 2: Spring Framework – Enterprise Application Development

- ✓ Introduction to the Spring Framework and Its Capabilities
- ✓ Exploring the Spring Framework Architecture
- ✓ **10 Key Reasons** to Use the Spring Framework in Java Projects
- ✓ **Core Concepts of Spring Framework**
- ✓ Understanding Core Spring and Its Features
- ✓ Overview of Key Spring Modules
- ✓ Inversion of Control (IoC) and Its Importance in Dependency Management

✓ Exploring IoC Containers in Spring:
- **Bean Factory** – Lightweight IoC Container
- **Application Context** – Advanced IoC Container for Enterprise Applications

✓ Working with Spring Beans and Their Configurations
✓ Understanding the Spring Bean Life Cycle and Its Stages
✓ Understanding Bean Scopes and Their Importance
✓ Configuring Spring Applications:
- **XML-Based Configuration**
- **Annotation-Based Configuration**
- **Java Code-Based Configuration**

Dependency Injection in Spring

✓ Understanding Dependency Injection and Its Benefits
✓ Implementing Dependency Injection:
- **Setter Injection**
- **Constructor Injection**

✓ Working with Autowiring in Spring
✓ Exploring Different Autowiring Modes
✓ Using the @Autowired Annotation for Dependency Resolution
✓ Leveraging the @Qualifier Annotation for Precise Bean Injection

Module 3: Spring Mvc Framework – Building Web Applications

✓ Introduction to the Spring MVC Framework
✓ Understanding the MVC (Model-View-Controller) Design Pattern
✓ Breaking Down the MVC Components:
- **Model** – Handling Business Logic and Data
- **View** – Presenting Data to Users
- **Controller** – Managing User Requests

Spring MVC Core Components

✓ Introduction to the Front Controller Pattern
✓ Role and Functionality of DispatcherServlet in Spring MVC
✓ Understanding the Request-Processing Flow in Spring MVC
✓ Key Advantages of Using Spring MVC

Building Web Applications with Spring MVC

- ✓ Using @Controller and @RequestMapping Annotations for Request Handling
- ✓ Creating Dynamic Views:
  - **JSP-Based Views**
  - **Thymeleaf-Based Views**
- ✓ Integrating Hibernate with Spring MVC for Database Operations

Module 4: Spring Boot Framework – Rapid Application Development

- ✓ Introduction to Spring Boot and Its Advantages
- ✓ Key Features of Spring Boot for Simplified Development
- ✓ Comparing **Spring Boot vs. Traditional Spring Framework**
- ✓ Understanding **Spring Boot Starter Web** for Web Applications
- ✓ Exploring Essential Spring Boot Annotations
- ✓ Using **Spring Boot CLI** for Quick Development
- ✓ Generating Projects with **Spring Initializr**
- ✓ Monitoring and Managing Applications with **Spring Boot Actuator**
- ✓ Introduction to **Spring Data JPA** for Database Integration
- ✓ Building a Complete Web Application Using Spring Boot

Module 5: Spring Security Framework – Securing Web Applications

- ✓ Introduction to Spring Security and Its Importance
- ✓ Real-World Use Cases of Spring Security in Applications
- ✓ **Core Concepts of Spring Security**
- ✓ Understanding the Spring Security Core Module
- ✓ Implementing **User Registration, Login, and Logout**
- ✓ Authentication Mechanisms in Spring Security
- ✓ Role-Based Authorization and Access Control
- ✓ Implementing **Password Encryption** for Secure Data Storage
- ✓ Using **JWT (JSON Web Tokens)** for Authentication

- ✓ Preventing Security Threats:
  - **CSRF (Cross-Site Request Forgery) Protection**
  - **CORS (Cross-Origin Resource Sharing) Management**

UNIT TESTING IN JAVA – ENSURING CODE RELIABILITY

- ✓ Understanding Unit Testing and Its Importance
- ✓ Using Assertions for Validating Test Results
- ✓ Introduction to Mocking and Method Stubbing
- ✓ Writing the First Unit Test Case
- ✓ Exploring Various Test Case Examples
- ✓ Creating Mocks and Stubs for Effective Testing
- ✓ Mocking a Class for Unit Testing
- ✓ Stubbing Methods to Simulate Behavior

WEB SERVICES – BUILDING RESTFUL APIs WITH SPRING BOOT

- ✓ Introduction to Web Services and Their Role in Modern Applications
- ✓ Different Types of Web Services:
  - **SOAP (Simple Object Access Protocol) Web Services**
  - **REST (Representational State Transfer) Web Services**

Developing RESTful APIs with Spring Boot

- ✓ Building a RESTful Web Service Using Spring Boot
- ✓ Implementing CRUD Operations:
  - **POST API** – Creating Resources
  - **GET API** – Retrieving Resources
  - **PUT API** – Updating Resources
  - **DELETE API** – Removing Resources
  - **PATCH API** – Partial Updates

Handling API Responses and Formats

- ✓ Understanding HTTP Response Status Codes
- ✓ Using the ResponseEntity Class for Custom API Responses
- ✓ Working with **JSON (JavaScript Object Notation)** and **XML** Data Formats

REST API Development with Spring Boot

- ✓ Creating and Configuring a REST Controller

- ✓ Understanding and Using REST Annotations:
  - @RestController – Defining REST Controllers
  - @RequestBody and @ResponseBody – Handling Request and Response Data
  - @PathVariable and @RequestParam – Extracting Request Parameters

Advanced REST API Features

- ✓ Using **DTO (Data Transfer Objects) and Mapper Classes** for Efficient Data Handling
- ✓ Implementing **Back-End Validation** with Annotations
- ✓ Handling Global Exceptions with @RestControllerAdvice
- ✓ Managing Application-Level Exception Handling
- ✓ Documenting REST APIs with **Swagger** for Better API Exploration and Testing

Web Designing:

HTML

Module 1: Introduction To Web Programming

- ✓ Understanding Website Architecture and Its Components
- ✓ Overview of the Internet and DNS (Domain Name System)
- ✓ Character Encoding: **UTF-8 vs. UTF-16**
- ✓ Importance of Meta Tags in Web Development
- ✓ Exploring Different Technologies Used in Web Development
- ✓ Introduction to Web Development and Its Core Concepts

Module 2: Fundamentals Of HTML

- ✓ Evolution and History of HTML
- ✓ Setting Up Your Development Environment
- ✓ Creating Your First HTML Page
- ✓ Understanding HTML Tags and Attributes
- ✓ Difference Between HTML Tags and Elements

- ✓ Exploring HTML Attributes and Their Usage

Module 3: HTML Head Elements

- ✓ Understanding the <head> Section of an HTML Document
- ✓ Importance and Usage of:
  - **<title>** – Defining the Page Title
  - **<base>** – Setting Base URLs for Relative Links
  - **<link>** – Linking External Stylesheets and Resources
  - **<style>** – Embedding Internal CSS Styles
  - **<script>** – Including JavaScript in Web Pages
  - **<meta>** – Providing Metadata for SEO and Browser Compatibility

Module 4: HTML Text Formatting And Structuring

- ✓ Exploring Basic HTML Tags for Structuring Content
- ✓ Using HTML Formatting Tags for Styling and Emphasis

Module 5: Grouping Elemnts in HTML Using And

- ✓ Understanding the Purpose of the Tag for Layout Structuring
- ✓ Using the Tag for Inline Grouping and Styling

Module 6: HTML Semantic Elements – Structured Content Markup

- ✓ Introduction to Semantic HTML and Its Importance in Web Development
- ✓ Exploring Key Semantic Elements:
  - **<article>** – Independent Content Blocks
  - **<aside>** – Sidebar Content and Related Information
  - **<details>** – Expandable Content Sections
  - **<figure>** – Associating Images with Captions
  - **<footer>** – Defining the Page or Section Footer
  - **<header>** – Representing the Header Section
  - **<main>** – Highlighting the Primary Page Content
  - **<mark>** – Highlighting Important Text
  - **<nav>** – Structuring Navigation Links

- **<section>** – Organizing Content into Sections
- **<summary>** – Providing a Summary for <details> Elements
- **<time>** – Representing Date and Time Values

Module 7: Creating Lists In HTML

- ✓ Understanding and Implementing Lists in HTML
- ✓ Creating Different Types of Lists:
  - **Unordered Lists (<ul>)** – Bullet Point Lists
  - **Ordered Lists (<ol>)** – Numbered or Alphabetized Lists
  - **Definition Lists (<dl>)** – Term and Description Lists

Module 8: Working With Images In HTML

- ✓ Embedding Images Using the <img> Tag
- ✓ Implementing Image Mapping for Interactive Images
- ✓ Setting and Managing Background Images in Web Pages

Module 9: Creating Hyperlinks In HTML

- ✓ Using the <a> (Anchor) Tag to Create Links
- ✓ Understanding **URLs (Uniform Resource Locator)** and Their Role
- ✓ Difference Between **Relative Path vs. Absolute Path** in Web Navigation

Module 10: Creating And Formatting Tables In HTML

- ✓ Understanding the Structure of an HTML Table
- ✓ Key Table Elements:
  - **<table>** – Creating a Table
  - **<th>** – Defining Table Headers
  - **<tr>** – Creating Table Rows
  - **<td>** – Adding Data Cells
  - **<caption>** – Providing Table Titles
  - **<thead>** – Grouping Table Headers
  - **<tbody>** – Organizing Table Body Content
  - **<tfoot>** – Defining Footer Rows in Tables

- **<colgroup>** – Grouping Table Columns
- **<col>** – Specifying Column Properties

Module 11: Embedding Content With HTML IFRAME

- ✓ Using <iframe> to Embed External Content
- ✓ Setting <iframe> as a Target for Links and Navigation

Module 12: Building Forms In HTML

- ✓ Introduction to HTML Forms and Their Purpose
- ✓ Key HTML Form Attributes and Their Functions
- ✓ Essential Form Elements for User Input:
- ✓ Understanding **HTML Input Types**
- ✓ Exploring **HTML Input Attributes**
- ✓ Utilizing **HTML Form-Specific Attributes**
- ✓ Additional Form Elements and Their Usage:
  - **<textarea>** – Creating Multi-Line Text Fields
  - **<button>** – Adding Clickable Buttons
  - **<select>** – Implementing Dropdown Lists
  - **<label>** – Associating Labels with Form Inputs

Module 13: HTML Graphics – Creating Visual Elements

- ✓ Using **HTML Canvas** for Dynamic Graphics
- ✓ Exploring **SVG (Scalable Vector Graphics)** for High-Quality Visuals

Module 14: Working With Media In HTML & Introduction To Github

HTML Media Elements

- ✓ Understanding Media Integration in HTML
- ✓ Embedding and Controlling **HTML Video**
- ✓ Implementing **HTML Audio** for Sound Playback
- ✓ Using **HTML Plug-ins** for External Media Support
- ✓ Integrating **YouTube Videos** into Web Pages

Version Control with Git & GitHub

- ✓ Setting Up a Real-Time Development Environment with GitHub
- ✓ Understanding **Git & GitHub** – Key Differences and Use Cases
- ✓ How Git Works – Basic Concepts and Workflow
- ✓ Practical Uses of Git & GitHub in Web Development
- ✓ Maintaining and Managing a GitHub Profile
- ✓ Deploying Web Pages Directly from GitHub

CSS

Module 15: Introduction To CSS – Styling The Web

- ✓ What Is **CSS (Cascading Style Sheets)?**
- ✓ Importance and Benefits of Using CSS
- ✓ Understanding CSS Syntax and Structure

Module 16: Core CSS Properties – Layout & Design

- ✓ **CSS Colors** – Applying and Managing Colors in Web Design
- ✓ **CSS Backgrounds** – Customizing Backgrounds for Elements
- ✓ **CSS Box Model** – Understanding Element Spacing and Dimensions
- ✓ **CSS Borders** – Adding and Styling Borders
- ✓ **CSS Margins** – Adjusting Outer Spacing of Elements
- ✓ **CSS Paddings** – Managing Inner Spacing of Elements
- ✓ **CSS Box Sizing** – Controlling Element Size Calculation

Module 17: CSS Units And Dimensions

- ✓ Understanding **CSS Units** for Sizing Elements
  - **Absolute Units** – Fixed-Size Measurements (px, cm, mm, in, pt)
  - **Relative Units** – Scalable Measurements Based on Context (em, rem, vw, vh, %)
- ✓ Managing Element Width with:
  - **Max-Width** – Setting Maximum Element Size
  - **Min-Width** – Defining Minimum Element Size

Module 18: Text And Font Styling In CSS

- ✓ Customizing **Text Appearance** (Color, Spacing, Decoration)
- ✓ Working with **CSS Fonts** (Custom Fonts, Font Families, Font Sizes)
- ✓ Understanding **CSS Outlines** for Element Highlighting
- ✓ Managing **Text Alignment** for Better Readability
- ✓ Using the !important Rule to Override Styles

Module 19: Styling HTML Elements With CSS

- ✓ Styling **Links** with Hover, Active, and Visited States
- ✓ Customizing **Lists** (Ordered, Unordered, Definition)
- ✓ Designing **Dropdown Menus** for Navigation
- ✓ Formatting **Tables** for Better Readability
- ✓ Styling **Images** Using:
  - **Image Sprites** for Optimized Performance
  - **Image Filters** for Visual Effects
  - **Clip-Path** for Custom Image Shapes
- ✓ Enhancing **Forms** for Better User Experience

Module 20: Creating Navigation Bars With CSS

- ✓ Designing **Vertical Navigation Bars**
- ✓ Implementing **Horizontal Navigation Bars**

Module 21: CSS Selectors – Targeting Elements Effectively

- ✓ **Simple Selectors** – Element, ID, and Class Selectors
- ✓ **Combinator Selectors** – Descendant, Child, Adjacent, and Sibling Selectors
- ✓ **Pseudo-Classes & Pseudo-Elements** – Styling Dynamic and Specific Parts of Elements
- ✓ **Attribute Selectors** – Selecting Elements Based on Attributes

Module 22: CSS Positioning And Flow Control

- ✓ Understanding **CSS Positioning** (Static, Relative, Absolute, Fixed, Sticky)
- ✓ Managing **Z-Index** for Overlapping Elements
- ✓ Controlling Element Flow with **Float & Clear**
- ✓ Handling Content Overflow with **Overflow Properties**

Module 23: CSS Layout Techniques

- ✓ Using the **Display Property** to Control Element Visibility and Behavior:
  - **Block, Inline, Inline-Block, None**
- ✓ Building Responsive Layouts with:
  - **CSS Grid** – Two-Dimensional Layouts
  - **CSS Flexbox** – Flexible, One-Dimensional Layouts

Module 24: Creating Gradient Effects With CSS

- ✓ **Linear Gradients** – Smooth Color Transitions in a Straight Line
- ✓ **Radial Gradients** – Circular Color Transitions
- ✓ **Conic Gradients** – Angular Color Transitions

Module 25: CSS Transformations And Animations

- ✓ **CSS 2D & 3D Transforms** – Scaling, Rotating, and Skewing Elements
- ✓ **CSS Transitions** – Smooth Property Changes Over Time
- ✓ **CSS Animations** – Creating Keyframe-Based Motion Effects

Module 26: Responsive Design With CSS

- ✓ Introduction to **Responsive Web Design**
- ✓ Understanding the **Viewport** for Mobile-Friendly Layouts
- ✓ Implementing **Responsive Breakpoints** for Different Screen Sizes
- ✓ Using **Media Queries** to Adapt Styles Based on Device Width

JAVASCRIPT

Module 1: Introduction To Javascript

- ✓ Understanding **JavaScript** and Its Role in Web Development
- ✓ Key Reasons to Use JavaScript in Modern Applications
- ✓ JavaScript **Syntax** and Structure
- ✓ Writing **Statements** for Program Execution
- ✓ Using **Comments** for Code Documentation

Module 2: Javascript Language Fundamentals

- ✓ Exploring **Data Types** in JavaScript
- ✓ Declaring Variables: **var vs. let vs. const**
- ✓ Understanding **Dynamic Typing** in JavaScript
- ✓ Using the typeof Operator for Type Identification
- ✓ Converting Between Data Types (**Type Conversion**)
- ✓ Working with **Objects** and Their Properties
- ✓ Manipulating **Arrays** for Data Storage

Module 3: Operators In Javascript

- ✓ Overview of **JavaScript Operators** and Their Functions
- ✓ Performing Calculations with **Arithmetic Operators**
- ✓ Assigning Values Using **Assignment Operators**
- ✓ Comparing Values with **Comparison Operators**
- ✓ Understanding **Equality Operators** (== vs. ===)
- ✓ Using **Ternary Operators** for Conditional Assignments
- ✓ Implementing **Logical Operators** for Decision Making
- ✓ Performing Bitwise Operations with **Bitwise Operators**

Module 4: Control Flow In Javascript

- ✓ Implementing Conditional Statements:
  - **if Statements** – Executing Code Based on Conditions
  - **else Statements** – Providing Alternative Code Execution
  - **if/else Statements** – Handling Multiple Conditions
  - **switch/case Statements** – Managing Multiple Conditional

Module 1: Looping Statements In Javascript

- ✓ Understanding **Loops** and Their Importance in Iteration
- ✓ Implementing Different Types of Loops:
  - **for Loop** – Basic Iteration Structure
  - **for/in Loop** – Iterating Over Object Properties
  - **for/of Loop** – Iterating Over Iterable Objects
  - **while Loop** – Executing Code While a Condition is True
  - **do/while Loop** – Ensuring Execution Before Condition Checking
- ✓ Handling **Infinite Loops** and Avoiding Pitfalls
- ✓ Using **Break and Continue** for Controlling Loop Execution

Module 1: Working With The Javascript HTML Dom

- ✓ Understanding the **HTML Document Object Model (DOM)**
- ✓ Exploring **DOM Methods** for Manipulating Web Pages
- ✓ Accessing and Modifying **DOM Documents and Elements**
- ✓ Handling **DOM HTML, Forms, and CSS Styling**
- ✓ Managing **DOM Events and EventListeners** for Interactivity
- ✓ Navigating the **DOM Tree** Using Different Methods
- ✓ Working with **DOM Nodes, Collections, and Node Lists**

Module 1: Working With Arrays In Javascript

- ✓ Introduction to **Arrays** and Their Structure
- ✓ Performing Basic Operations:
  - **Adding, Deleting, and Modifying Elements**
  - **Searching and Emptying Arrays**
  - **Combining and Slicing Arrays**
- ✓ Utilizing the **Spread Operator** for Array Expansion
- ✓ Iterating Over Arrays with Different Techniques:
  - **Looping, Filtering, Mapping, and Reducing Arrays**

Module 1: Javascript Functions And Scope

- ✓ Declaring and Using **Functions** Effectively

- ✓ Understanding **Function Hoisting**
- ✓ Handling Function **Arguments and the Rest Operator (...args)**
- ✓ Setting **Default Parameters** in Function Calls
- ✓ Using **Getters and Setters** for Encapsulation
- ✓ Implementing **Try and Catch** for Error Handling
- ✓ Exploring **Local vs. Global Scope** in JavaScript
- ✓ Understanding the **this Keyword** and Its Context

Module 5: Javascript Objects And Data Handling

- ✓ Basics of **Objects** in JavaScript
- ✓ Creating Objects Using **Factory Functions and Constructors**
- ✓ Understanding the **Dynamic Nature of Objects**
- ✓ Treating **Functions as Objects** in JavaScript
- ✓ Exploring **Value vs. Reference Type** Variables
- ✓ Enumerating and Managing **Object Properties**
- ✓ Cloning Objects for Data Duplication
- ✓ Working with **String and Date Objects**

Module 6: Javascript Built-in Objects

- ✓ Exploring Core JavaScript Objects:
  - **Number** – Handling Numeric Values
  - **Math** – Performing Mathematical Operations
  - **String** – Manipulating Text Data
  - **Array** – Managing Collections of Data
  - **Date** – Working with Dates and Time
  - **Boolean** – Handling True/False Logic
  - **Regex** – Using Regular Expressions for Pattern Matching

Module 7: Javascript Event Handling

- ✓ Introduction to **Events** and Their Importance
- ✓ Handling Different Types of Events:
  - **Mouse Events** – Click, Hover, Drag
  - **Keyboard Events** – Key Presses and Shortcuts

- **Form Events** – Input, Submit, Change
- **Document & Window Events** – Page Load, Resize, Scroll

Module 8: Introduction To Object-oriented Programming (Oop) In Javascript

- ✓ Understanding **Prototypes and the Module Pattern**
- ✓ Augmenting Data Types and Using **Closures**
- ✓ Introduction to **ES6 Features** for Modern JavaScript Development
- ✓ Exploring **Let & Const** for Variable Declaration
- ✓ Using **Arrow Functions** for Concise Syntax
- ✓ Working with **Classes and Inheritance** in JavaScript
- ✓ Understanding **Rest and Map Operators**
- ✓ Managing **Exports and Imports** for Modular JavaScript Code

Module 9: Javascript Web Api Integration

- ✓ Introduction to **Web APIs** and Their Role
- ✓ Working with Different APIs:
  - **Forms API** – Handling Form Data Efficiently
  - **History API** – Managing Browser History
  - **Storage API** – Local and Session Storage
  - **Worker API** – Running Background Scripts
  - **Fetch API** – Fetching Data from Servers
  - **Geolocation API** – Accessing User Location

Module 10: Ajax And Asynchronous Javascript

- ✓ Introduction to **AJAX (Asynchronous JavaScript and XML)**
- ✓ Understanding **AJAX Requests and Responses**
- ✓ Implementing **AJAX with XMLHttpRequest**
- ✓ Handling **AJAX Requests** for Fetching and Sending Data
- ✓ Managing **AJAX Responses** for Real-Time Web Interactions

Module 11: Asynchronous Javascript And Ajax

- ✓ Introduction to **AJAX (Asynchronous JavaScript and XML)**

- ✓ Understanding **XMLHttpRequest (XHR) API** for Asynchronous Data Handling
- ✓ Making an **AJAX Request** to Fetch Data from a Server
- ✓ Processing and Handling an **AJAX Response** Efficiently

Module 12: Working With Json (Javascript Object Notation)

- ✓ Introduction to **JSON (JavaScript Object Notation)** and Its Uses
- ✓ Understanding **JSON Syntax and Structure**
- ✓ Exploring Different **JSON Data Types**
- ✓ Parsing JSON Data with **JSON.parse()**
- ✓ Converting JavaScript Objects to JSON Strings with **JSON.stringify()**
- ✓ Working with **JSON Objects and Arrays** for Data Management

JQUERY

Module 1: Mastering JQUERY For Dynamic Web Development

- ✓ **Getting Started with jQuery** – Introduction and Setup
- ✓ **Selecting Elements** Efficiently with jQuery Selectors
- ✓ **Manipulating the DOM** – Modifying Page Content Dynamically
- ✓ **Traversing the DOM** and Utilizing Chaining for Efficiency
- ✓ **Using jQuery Utility Methods** for Simplified Development
- ✓ **Handling Events and Event Delegation** for Dynamic Interactions
- ✓ **Working with AJAX, JSON, and Deferred Objects** for Asynchronous Operations
- ✓ **Enhancing User Experience with Animations and Effects**
- ✓ **Implementing Grids and Tables with AJAX, Pagination, and jQuery UI**
- ✓ **Following jQuery Best Practices** for Performance Optimization and Maintainability

BOOTSTRAP

Module 1: Introduction To Bootstrap Framework

- ✓ Overview and Introduction to Bootstrap
- ✓ Exploring **Quick Styles After Dark** for Rapid Prototyping

- ✓ Introduction to Bootstrap UI Components
- ✓ Creating Interactive **Dropdown Menus**
- ✓ Implementing **Buttons with Dropdown Menus**
- ✓ Designing **Tabs & Pills** for Navigation
- ✓ Customizing **Navbars** for Seamless User Experience

Module 2: Building Layouts With Bootstrap

- ✓ Introduction to Bootstrap's Layout System
- ✓ Working with **Grid Layouts** for Flexible Design
- ✓ Creating **Simple Layouts** for Web Pages
- ✓ Implementing **Fixed Grids** for Structured Content
- ✓ Utilizing **Fluid Grids** for Adaptive Sizing
- ✓ Understanding **Responsive Design** Principles
- ✓ Using **Responsive Utilities** for Mobile Optimization
- ✓ Finding **Design Inspiration** for Modern Web Development
- ✓ **Summary and Best Practices**

Module 3: Essential Bootstrap Components

- ✓ Introduction to Everyday Bootstrap Elements
- ✓ Styling and Formatting with **Typography**
- ✓ Creating and Managing **Tables**
- ✓ Designing User-Friendly **Forms**
- ✓ Customizing **Buttons** for Interaction
- ✓ Enhancing Visuals with **Images & Icons**
- ✓ **Summary and Key Learnings**

Module 4: Bootstrap Components – Enhancing Ui Elements

- ✓ Creating **Our First Web Page** with Bootstrap
- ✓ Understanding **Page Structure and Components**
- ✓ Managing **Scripts & Styles** in Bootstrap
- ✓ Core Concepts of Bootstrap for Efficient Development
- ✓ **Summary and Key Takeaways**
- ✓ Enhancing Web Pages with **Heroes, Badges, Labels, and Media Elements**
- ✓ **Summary and Best Practices**

Module 5: Bootstrap With Javascript – Interactivity & Dynamics

- ✓ Introduction to JavaScript-Enabled Bootstrap Components
- ✓ Exploring **Advanced Button Functionalities**
- ✓ Implementing **Modals** for Dialogs and Popups
- ✓ Adding **Tooltips and Popovers** for Better User Interaction
- ✓ Creating Engaging **Carousels** for Image and Content Slideshows
- ✓ **Summary and Key Takeaways**

Learning Management System

Key Features Learning Management System

Live Sessions with Class Recordings

Gain access to interactive live training sessions along with recorded classes to reinforce learning at your own pace.

Expert-Led Training Sessions

Learn from industry experts through structured training sessions designed to enhance your skills and practical knowledge.

Earn Your Training Certificate

Receive an official training certification upon successful course completion to validate your learning and expertise.

Experience Certificate for Professionals

Get an experience certificate based on your hands-on project work and practical assessments.