# NetTech India

Get Certified



## Certified Professional Diploma in

# FULL STACK JAVA

- Practical Training
- Training From Expert Trainer
- Interview Preparation
- Complete Placement Assistance

https://www.nettechindia.com

info@nettechindia.com

# CERTIFICATIONS OPTIONS AVAILABLE









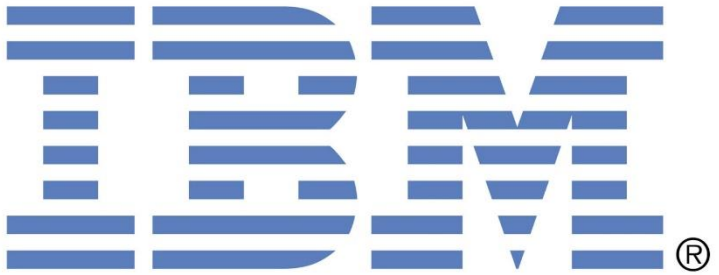










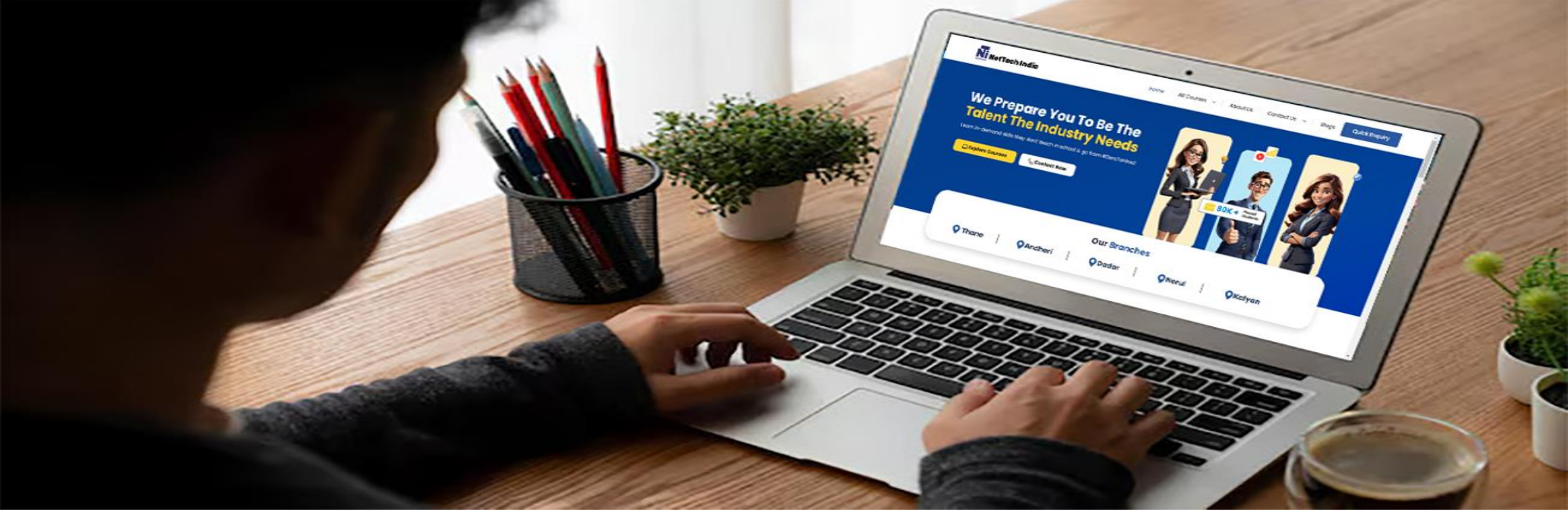





Microsoft




NetTech India




# ABOUT US

**NetTech India** is a premier training institute specializing in cutting-edge, job-ready courses in the field of IT, CAD & Accounts. With a focus on practical, hands-on learning, we offer industry-recognized certifications and personalized training programs. Our expert instructors are committed to helping students and professionals enhance their skills and advance their careers. Located in **Thane**, **Mumbai** and **Navi Mumbai**, NetTech India is dedicated to providing quality education and fostering a culture of continuous learning and innovation. We Prepare You To Be The Talent The Industry Needs!

**25%**
Theory

**75%**
Practicals

# ABOUT FULL STACK JAVA

Full Stack Java development involves building both the front-end and back-end of web applications using Java technologies. It’s one of the most in-demand skill sets in the software industry today. This course provides complete training in Web Designing using HTML, CSS, JavaScript, and JQuery, along with modern front-end development using React JS. On the server side, students gain strong expertise in Core Java, Advanced Java, Java Frameworks, and Spring Boot. By mastering these tools, learners can create dynamic, responsive, and scalable web applications, making them highly valuable and job-ready in today’s tech-driven world.



# BENEFITS OF FULL STACK JAVA

- Career Growth - Higher Pay & Position
- Encourages professional Development
- Enriches self-image and Reputation
- Enhances professional Credibility.
- Abundant Job Opportunities
- Used In Many Industries
- Global Recognition
- Secure and Flexible
- 50+ Case Studies
- 10+ Projects



# FULL STACK JAVA TOPICS

- **WEB DESIGNING (HTML, CSS, JAVASCRIPT & JQUERY)**
- **REACT JS**
- **CORE JAVA**
- **ADVANCED JAVA**
- **JAVA FRAMEWORK**

# OBJECTIVES

With this progressive HTML5 ,CSS3, JAVA SCRIPT and J-QUERY combo course, we brings forth yet another blended training module comprising of two of the most popular and rapidly emerging technologies. HTML5 and CSS3 is the next big thing in the world of web development. HTML5 has risen to the status of a core markup language in the world wide web due to a number of new attributes and elements that are apt for use in modern websites.

This training will also teach you how to: code application logic in web applications with the use of JavaScript, store local data using JSON, create HTML5 pages to parse data with the use of HTML5 forms, build the DOM, build responsive layouts with CSS3, and describe on HTML5 canvas.

# HTML

**Overview of HTML5:**

- Introduction HTML
- HTML Basics HTML
- Structure of HTML5
- Different between HTML and HTML5
- Elements HTML5
- Semantic HTML Attributes and Formatting Tags
- HTML Headings
- HTML Paragraph
- HTML Pre tag
- HTML Mark Tag
- HTML br and hr tag
- HTML Image with alt
- HTML Styles
- HTML Formatting
- HTML Quotations
- HTML Computer Code
- HTML Comments & Colours HTML CSS,
- Links and Images
- HTML Lists

- HTML Table
- HTML Blocks
- HTML SVG
- HTML Classes
- HTML iframe - Google Map
- HTML Layout
- HTML Responsive
- HTML JavaScript
- HTML Forms - Labels and Inputs , Radio , Date , Time , Checkbox, Select, Textarea, Button-sumbit , Input-Required , Max- length and Min- length
- Html Marquee
- Html variables
- Figure Tag, Video,Audio

# CSS

- Introduction CSS3
- CSS3 Syntax
- CSS3 How To link CSS in html
- CSS3 Colours
- CSS3 Backgrounds
- CSS3 Background Image
- CSS3 Borders
- CSS Padding
- CSS Margin
- CSS Calc method
- CSS hover
- CSS importants
- CSS opacity
- CSS scale
- CSS skew
- CSS rotate c
- CSS3 Font Style
- CSS Height/Width
- CSS Animation - Keyframes
- CSS3 Gradients
- CSS3 Position Fixed and Sticky

- CSS3 Shadows
- CSS3 Text
- CSS3 Fonts
- CSS3 2D Transforms
- CSS3 3D Transforms
- CSS Lists, CSS Tables
- CSS Outline
- CSS Display
- CSS Max-width
- CSS Position
- CSS Justify Contant
- CSS Float
- CSS Inline-block
- CSS Align
- CSS class
- CSS element
- CSS Navigation Bar
- CSS Dropdowns
- CSS3 Images
- CSS Attribute
- Selectors CSS
- Forms CSS Counters
- CSS3 Animations

- CSS3 Buttons
- CSS3 Pagination
- CSS3 Multiple Columns
- Border Image
- Media Queries, CSS3 Responsive

# JAVA SCRIPT

- JavaScript Introduction,
- Internal, External  javascript
- Data Types
- Js Operators
- If Statements
- Switch,Loops Js
- Function
- Function return
- Function with parameters
- Arrow Functions
- Calculator Projects using Function
- Object
- Arrays Js
- Array Methods
- Array Map, Filter, Reduce
- Maths Methods
- Try and catch
- DOM(Document Object) getElementById, GetElementsByName, getElementsByTagName, InnerHtml, InnerText property Js
- Validation – Form validation, Email validation Js

# JQUERY

- Introduction to jQuery
- JQuery Syntax
- JQuery Selectors
- JQuery Events
- Hide, Show, Toggle,
- FadeIn, FadeOut,
- SlideDown, SlideUp, SlideToggle,
- Jquary animate
- JQuery Mouseover and Mouseleave
- JQuery Keypress, Keyup and Keydown
- JQuery Focus, blur, select, submit
- JQuery Click and dblclick
- JQuery CSS
- JQuery Effects
- JQuery HTML
- JQuery Append, prepend, before, after

# REACT JS

## REACTJS – OVERVIEW

- React – Features
- React – Advantages
- React – Limitations

## REACTJS – ENVIRONMENT SETUP

- Install Global Packages
- Create the Root Folder
- Add Dependencies and Plugins
- Create the Files
- Set Compiler, Server, and Loaders
- index.html
- App.jsx and main.js
- Running the Server

## REACTJS – JSX

- Using JSX
- Nested Elements
- Attributes
- JavaScript Expressions
- Styling

- Comments
- Naming Convention

**REACTJS – COMPONENTS**

- Stateless Example
- Stateful Example

**REACTJS – STATE, EFFECT & REF**

**REACTJS – PROPS OVERVIEW**

- Using Props27 Default Props
- State and Props

**REACTJS – PROPS VALIDATION**

- Validating Props

**REACTJS – COMPONENT API**

- Set State
- Force Update
- Find Dom Node

**REACTJS – COMPONENT LIFE CYCLE**

- Lifecycle Methods

## REACTJS – FORMS

- Simple Example
- Complex Example

## REACTJS – EVENTS

- Simple Example
- Child Events

## REACTJS – ROUTER

- Install a React Router
- Create Components
- Add a Router

## FETCH API

- Then
- Catch

## REACTJS – ANIMATIONS

- Install React CSS Transitions Group
- Add a CSS file
- Appear Animation
- Enter and Leave Animations

**REACTJS – HIGHER ORDER COMPONENTS**

**REACTJS – BEST PRACTICES**

**Node JS – CRUD Operations**

**And Many More...**

# JAVA

## CORE JAVA

### Java Basics

- Define the scope of variables. Define the structure of a Java class. Create executable Java applications with a main method; run a Java program from the command line; Import other Java packages to make them accessible in your code. Compare and contrast the features and components of Java such as: platform independence, object orientation, encapsulation

### Working with Java Data Types

- Declare and initialize variables (including casting of primitive data types). Differentiate between object reference variables and primitive variables. Develop code that uses wrapper classes

### Working with Operators and Decision Constructs

- Using Operators and Decision Constructs. Use Java operators; including parentheses to override operator precedence. Test equality between Strings and other objects using == and equals (). Create if and if/else and ternary constructs .Use a switch statement

## Creating and Using Arrays

- Declare instantiate, initialize and use a one-dimensional array: Declare, instantiate, initialize and use multi-dimensional array.

## Using Loop Constructs

- Create and use while loops. Create and use for loops including the enhanced for loop. Create and use do/while loops. Compare loop constructs. Use break and continue

## Java Class Design

- Implement encapsulation. Implement inheritance including visibility modifiers and composition. Implement polymorphism. Object class. Create and use singleton classes and immutable classes. static keyword on initialize blocks, variables, methods, and classes

## Working with Methods and Encapsulation

- Create methods with arguments and return values. Including overloaded methods. Apply the static keyword to methods and fields. Create and overload constructors; including impact on default constructors. Apply access modifiers

## Working with Inheritance

- Describe inheritance and its benefits. Develop code that demonstrates the use of polymorphism; including overriding and object type versus reference type. Determine when casting is necessary. Use super and this to access objects and constructors. Use abstract classes and interfaces

## Handling Exceptions

- Differentiate among checked exceptions, unchecked exceptions, and Errors. Create a try-catch block and determine how exceptions alter normal program flow. Describe the advantages of Exception handling. Create and invoke a method that throws an exception. "Recognize common exception classes (such as NullPointerException, ArithmeticException, ArrayIndexOutOfBoundsException, ClassCastException)"

## Working with Selected classes from the Java API

- Manipulate data using the String Builder class and its methods. Creating and manipulating Strings. Create and manipulate calendar data using classes from java.time.LocalDateTime, java.time.LocalDate, java.time.LocalTime, java.time.format

- DateTimeFormatter, java.time.Period . Declare and use an ArrayList of a given type. Write a simple Lambda expression that consumes a Lambda Predicate expression

## Generics and Collections

- Create and use a generic class. Create and use ArrayList, TreeSet, TreeMap, and ArrayDeque objects. Use java.util.Comparator and java.lang.Comparable interfaces. Collections Streams and Filters.Iterate using forEach methods of Streams and List. Describe Stream interface and Stream pipeline. Filter a collection by using lambda expressions. Use method references with Streams

## Lambda Built-in Functional Interfaces

- Use the built-in interfaces included in the java.util. function package such as Predicate, Consumer, Function, and Supplier. Develop code that uses primitive versions of functional interfaces. Develop code that uses binary versions of functional interfaces. Develop code that uses the Unary Operator interface

## Java Stream AP

- Develop code to extract data from an object using peek() and map() methods including primitive versions of the map() method. Search for data by using search methods of the Stream classes including findFirst, findAny, anyMatch, allMatch, noneMatch. Develop code that uses the Optional class. Develop code that uses Stream data methods and calculation methods. Sort a collection using Stream API. Save results to a collection using the collect method and group/partition data using the Collectors class. Use flatMap() methods in the Stream API

## Use Java SE 8 Date/Time API

- Create and manage date-based and time-based events including a combination of date and time into a single object using LocalDate, LocalTime, LocalDateTime, Instant, Period, and Duration. Work with dates and times across time zones and manage changes resulting from daylight savings including Format date and times values.

## Java I/O Fundamentals

- Read and write data from the console. Use BufferedReader, BufferedWriter, File, FileReader, FileWriter, FileInputStream, FileOutputStream, ObjectOutputStream, ObjectInputStream, and PrintWriter in the java.iopackage

# ADVANCED JAVA

**JDBC**

- Introduction to JDBC. JDBC architecture. java. SQL Package
- Connection, Statement, Result Set
- Prepared Statement.
- Result Set. Batch Updates.
- Result Set Metadata.
- Simple Transaction Management

**SERVLETS**

- The need for server-side programming. Introduction to Servlets. Servlet Life Cycle
- Javax .servlet package. Servlet Config, Servlet Context, Servlet Response Supplying initialization parameters to Servlets
- Performing database operations in Servlets
- Include and forward mechanisms
- Applying filters to Servlets

- java. servlet. http Package. Http Servlet Life Cycle. Http request methods GET vs. POST. Http Servlet
- Request, Http Servlet Response. Dealing with Http headers & error codes
- Session Tracking, purpose .Hidden form fields, Cookies .Http Session, URL rewriting

**JSP**

- Introduction to JSP. JSP Life Cycle
- Creating dynamic Web content with JSP
- Scripting elements: Scriptlet. Declaration. Expression
- XML syntax for JSP elements .JSP directives page, include and tag lib. JSP implicit objects. JSP scopes
- Include and forward mechanism
- Using a Java bean in a JSP. JSP Model 1 architecture. JSP Model 2 (MVC) architecture
- Custom Tag Development. Classic Tags, Simple Tags. Error Handling in a JSP
- JSTL. (Java Standard Tag Library)
- Expression Language.Processing XML in a JSP

## And Many More...

# JAVA FRAMEWORK

**HIBERNATE**

- Introduction to Hibernate
- Object Relational Mapping
- Hibernate Architecture
- Hibernate Mapping
- Collection in Hibernate
- Inheritance in Hibernate
- Hibernate Query Language(HQL)

## Spring

- Overview
- Architecture
- IOC Containers
- Bean Definition, Scope
- Dependency Injection with Bean
- Annotation Based Configuration
- XML Based Configuration
- MVC with Spring
- AOP with Spring Framework
- Spring JDBC Framework

## And Many More...

# WHO CAN LEARN?

- Anyone Who Wants To Build A Career In IT Field
- Anyone Who Wish To Gain Knowledge About Programming
- Students Who Are Currently In College Or University



# CAREER OPPORTUNITIES

- Frontend Web Developer
- Backend Web Developer
- UI/UX Developer
- Web Application Developer
- Full Stack Developer
- Spring & Spring boot Developer
- Software Developer
- Software Engineer

**And Many More...**



# OUR RECRUITERS




**AND MANY MORE.....**

# PROCESS FOR SUCCESS

GET PLACED

GET TRAINED

ENROLL



# FACILITIES OFFERED

- Practical Training On Live Projects
- Complete Placement Assistance
- Interview Preparation
- Global Certification
- Fully Functional Labs
- Online / Offline Training
- Study Materials
- Expert Level Industry Recognized  Training



**Note:** NetTech India's job placement assistance is contingent upon students attending and actively participating in the prescribed placement training sessions.

V 11.1

