NXT WAVE
INTENSIVE

NXT WAVE™

JAVA FULL STACK

Build a strong understanding of Java programming and web development technologies to become a Full-Stack Java Developer with React. Get hands-on experience with various Java frameworks for backend development and React for frontend development. Create a portfolio of projects by developing end-to-end web applications using Spring Boot and React frameworks.

NXT WAVE INTENSIVE

NXT WAVE™

HTML & CSS

In this course, you will learn how to design and implement layouts that adapt seamlessly across different screen sizes using techniques like CSS Flexbox, media queries, and Tailwind CSS. The course will guide you through structuring HTML pages and applying styles to make websites responsive and user-friendly.

Software Development Fundamentals

- Importance of Frontend
- Fundamentals
- Syntax
- Debugging
- Tweaking Code

HTML Basics

- Getting Started
- HTML Elements
- HTML Headings
- HTML Paragraphs
- HTML Button
- HTML Attributes
- HTML Images
- HTML Hyperlinks
- HTML Containers
- HTML Lists

CSS Basics

- CSS Rulesets
- CSS Properties
- CSS Colors
- CSS Units
- CSS Box Model
- CSS Margins
- CSS Paddings
- CSS Borders

CSS Layouting

CSS Flexbox

- Display
- Flex-direction
- Justify-content
- Align-items
- Flex-wrap
- Flex-row
- Align-content
- Align-self
- Flex-grow
- Flex-shrink
- Order

Sizing Elements

- Overflow
- Box Sizing
- Content Box
- Border Box

CSS Media Queries

- Media Query
- Media Types
- Media Features
- Width, min-width, max-width
- Height, min-height, max-height
- Orientation
- Combining Multiple Media Queries
  - Logical Operators

CSS Grid

- Implicit and Explicit Grid
- Grid rows and columns
- Grid Positioning and Alignment

CSS Positioning

- Position Property
  - Static, fixed, relative, absolute
- Changing Element's Position
  - Relative Positioning
  - Absolute Positioning
  - Fixed vs Absolute
- Positioned Ancestors

Tailwind CSS

- Introduction to Tailwind CSS
  - Why Tailwind CSS?
  - Advantages
  - Setting up Tailwind CSS
  - Common Utility Classes
- Tailwind Display & Responsive Utilities
  - Flex properties
  - Grid Utilities
  - Responsive Utilities

Course Project

E-commerce Website

Build a responsive e-commerce website using HTML and CSS

JavaScript

In this course, you will learn JavaScript concepts and programming techniques to build dynamic web applications. You'll master the core features of JavaScript, including DOM manipulation, event handling, arrays, functions, and working with real-time data through HTTP requests. By the end of this course, you'll be able to create interactive and responsive websites that respond to user actions and integrate with real-world data.

Introduction to JavaScript

- JavaScript Introduction
- ECMAScript
- Browser Console

Variables & Data Types in JavaScript

- Declaring Variables
- Understanding Primitive Types and the typeof Operator
- Variable Assignment and Naming Rules

Type Conversion

- Implicit and Explicit Conversion
- Converting Strings to Numbers and Booleans

Sequence of Instructions & Operators

- Understanding Sequence of Instructions and Expressions
- Working with Modulus, Exponentiation, Increment, and Decrement Operators
- BODMAS Rule in JavaScript

Operators

- Relational Operators
- Loose vs Strict Equality and Type Coercion
- Logical Operators
- Conditional Statements
- Nested Conditions and the Switch Statement
- Using the Ternary Operator

String Basics & Methods

- String Concatenation, Template Literals, and Indexing
- String Methods

Introduction to Loops

- while and do-while Loop Syntax
- The for Loop and Nested Loops
- Loop Control: break, continue

Functions

- Function Declaration, Parameters, and Arguments
- Understanding Return Statements and Default Parameters
- Named Function Expressions vs Function Declarations
- Callback Functions

Arrow Functions, Scope & Hoisting

- Arrow Function Syntax and Implicit Return
- Understanding Scope: Global, Function, and Block Scope
- Hoisting with var, let, const, and Functions

Arrays

- Creating Arrays and Accessing Array Elements
- Array Methods

Objects

- Working with Objects: Creating, Accessing, and Modifying Properties
- Iterating Over Objects
- Creating and Iterating Over Arrays of Objects
- Filtering, Mapping, and Sorting Arrays of Objects

Error Handling

- Using try-catch for Error Handling
- The finally Block

JavaScript Functions

- Creating Objects Using Factory Functions and Constructor Functions
- Understanding Object Property Value Shorthand Notations

JavaScript Classes

- Introduction to JS Classes and Constructor Methods
- Inheritance using extends and super()

JavaScript Promises

- Understanding Asynchronous vs Synchronous Code
- Promise States: Pending, Fulfilled, Rejected
- Consuming Promises with .then(), .catch(), and Promise Chaining
- Creating and Consuming Promises
- Using async and await for Simplified Asynchronous Code

Dynamic Javascript

In this course, you'll learn essential JavaScript concepts to build dynamic web applications. Focus areas include DOM manipulation, event handling, schedulers, and HTTP requests. By the end, you'll be able to create responsive websites that interact with users and real-time data.

DOM and Events

- DOM Manipulation
- Event Handling

Callbacks & Schedulers

- Callbacks: Functions as arguments
- Schedulers: setInterval( ), setTimeout( )

Event Listeners

- Event Types: keydown, keyup
- Event Object: event.type, event.target

HTTP Basics

- Protocols: HTTP, HTTPS
- Request/Response: Start Line, Headers, Body

HTTP Requests in JS

- Methods: GET, POST, PUT, DELETE
- JSON Stringify and Parsing

Forms

- Form Elements
- Form Events

JS Modules

- CommonJS Module Exports
- ES6 Module Exports

Node Packages

- Core Modules: Path, Package
- NPM: Creating Projects, Third-Party Packages

NXT WAVE INTENSIVE

NXT WAVE™

React JS

In this course, you will learn how to build stateful web applications with the ReactJS library. Upon completion, you will be comfortable creating an application in ReactJS from scratch.

Introduction to React JS

- Why React?
- Introducing JSX
- Rendering Elements, Components, and Props

State and Effect Hooks

- useState(), useEffect()
- Handling Events
- Component Life Cycle

Routing

- useParams(), useNavigate()
- Protected Route with JWT Authentication

React Context

- use() Provider Component

Course Project

E-commerce Application

Build an e-commerce application using React JS

Core Java

In this course, you will learn the fundamental concepts of Java programming. Upon completing this course, you will be able to write high-quality and reusable code in Java.

Java Environment & Basics

Objective: Build a strong foundation in Java and understand how programs run.

- How to Learn a New Programming Language in Record Time
- Introduction to Java
- JVM, JRE, JDK, History, Features
- Variables & Data Types
- Working With Keywords & Variables
- Scanning User Input
- Displaying Output in Java

Operators, Strings & Control Flow

Objective: Build a strong foundation in Java and understand how programs run.

- Operators | Part 1
- Operators | Part 2
- Working with Strings
- Working with String Methods
- String Handling in Java
- Decision Making (if/else, switch)
- Loop and Jump Statements

Object-Oriented Programming (OOP)

Objective: Build a strong foundation in Java and understand how programs run.

- Introduction to OOP
- Object-Oriented Programming | Part 2
- Methods | Part 1
- Methods | Part 2
- Method Call Stack & Recursion
- Constructors in Java
- StringBuilder & StringBuffer
- Inheritance
- Packages and Access Modifiers
- Polymorphism (Overloading & Overriding)
- Abstraction (Abstract Classes)
- Interface
- Functional Interface and Lambda Expression
- Encapsulation
- Equality & Ordering in Java

Arrays

Objective: Work with structured data efficiently.

- Arrays in Java | 1D
- Arrays in Java | 2D

Collections Framework

Objective: Master dynamic data structures and modern Java utilities.

- Generic Class & Collection Framework
- ArrayList
- LinkedList
- Iterator
- Comparable and Comparator
- Understanding Java Set Collections
- HashMap
- Operations in HashMap
- LinkedHashMap & TreeMap
- Stack
- Queue and Deque
- Mastering Java Stream API
- Java 8 Features: Method Reference & Optional

Exception Handling

Objective: Build robust and error-resistant applications.

- Exception Handling in Java
- Types of Exceptions
- Advanced Exception Handling
- throw vs throws
- Checked vs Unchecked Exceptions
- Custom Exceptions

Multithreading

Objective: Learn concurrency and performance optimization.

- MultiThreading in Java | Part 1
- MultiThreading in Java | Part 2
- Creating Threads (Thread, Runnable)
- Thread Lifecycle
- Synchronization
- Synchronization & Deadlock
- Inter-thread Communication
- Executor Framework

Core Java Advanced Concepts & Utilities

Objective: Understand important concepts of Enum, explore new features of Collections and Multithreading, and learn key concepts of File Handling.

- Enum in java
- Concurrent Collection in JAVA
- File Handling and Serialization in Java
- Byte Streams vs Character Streams

Introduction to Databases

Learn about the incredibly prevalent databases today. Through this course, you'll develop strong fundamentals and be proficient in concepts related to Databases and DBMS. Most importantly, you'll also be able to perform powerful queries on databases using SQL.

Relational Databases

- Introduction to SQL
- Database Design
- ACID Properties

SQL Operations

- CREATE
- INSERT
- SELECT
- UPDATE
- DELETE

Advanced SQL Modelling

- ER Model
- Creating a Relational Database

Advanced SQL

- Joins
- Subqueries
- Indexes
- Views

Advanced Java

In this course, you will deepen your knowledge of Java and web development. You will learn how to connect to databases, handle web requests, and implement a variety of Java technologies in real-world applications.

Database Connectivity (JDBC)

- JDBC Architecture
- JDBC Drivers
- Establishing Connection with SQLite
- Executing SQL (Statement, PreparedStatement)
- ResultSet.
- Trancation Management

Course Project

Mini Project: Employee Management System (JDBC)

A console-based application using JDBC to perform CRUD operations on employee data.
Features:

Features

- Add, View, Update, and Delete employee records

Learning Outcome

- Database interaction using JDBC
- CRUD operations implementation
- Executing SQL queries in Java

Servlet & JSP

- Web App Architecture & Servlet Basics
- Handling Requests & Responses
- Request Dispatch & URL Mapping
- Servlet Context & Config
- Session Management - I (Cookies)
- Session Management - II (HttpSession)
- JSP Architecture & Lifecycle
- JSP Directives, Scripting & Implicit Objects
- EL & JSTL
- MVC Pattern + Database Integration

Course Project

A web-based application built using Servlet, JSP, and JDBC following the MVC architecture.
Features:

Features

- User Login & Session Management
- Add, View, Update, Delete Users (CRUD)
- Data display using JSP (EL & JSTL)

Learning Outcome

- End-to-end web application flow (Request → Servlet → DB → JSP)

Mini Project: User Management System (Servlet, JSP & JDBC)

- Session & Cookie handling
- MVC design implementation

Hibernate

In this course, you will learn ORM concepts, JPA, and Hibernate to build database-driven applications, including configuration, entity mapping, lifecycle management, CRUD operations, transactions, relationships, and HQL queries.

Module 1: ORM & JPA Fundamentals

- Limitations of JDBC
- Boilerplate code
- Manual object–table mapping
- Introduction to ORM (Object Relational Mapping)
- Introduction to JPA (Java Persistence API)
- Hibernate as JPA Implementation
- Benefits of using Hibernate

Module 2: Hibernate Architecture & Configuration

- Hibernate Architecture Overview
- Configuration
- SessionFactory
- Session
- Transaction
- Hibernate Configuration (hibernate.cfg.xml)
- Database Connectivity Setup
- Application Flow: Application → Hibernate → Database

Module 3: Entity & Persistence Lifecycle

- Entity Class Basics
- @Entity, @Id, @Table
- Persistent Class Concept
- Object Lifecycle States
- Transient
- Persistent
- Detached

Module 4: CRUD Operations & Transaction Management

- Insert Operation (save)
- Fetch Operation (get, load)
- Update Operation
- Delete Operation
- Transaction Management
- Introduction to First-Level Cache (Session-level caching)

Module 5: Mapping, Relationships & Query Language

- Mapping
- @Column
- Basic Data Types
- Relationships
- One-to-One
- One-to-Many
- Many-to-One
- HQL (Hibernate Query Language)
- Basic Queries (from, where)

NXT WAVE INTENSIVE

NXT WAVE™

Enterprise Java Development with Spring , Spring Boot & Microservices

In this course, you will Learn to build scalable, secure, and database-driven backend applications using Spring Core, Spring Boot, JPA, REST APIs, Security (JWT), and Microservices architecture.

1. SPRING CORE & IOC (Foundation)

- Spring Framework Fundamentals
- Inversion of Control (IoC)
- Dependency Injection (DI)
- Spring Beans — Definition & Scopes
- Annotation-Based Configuration (focus more)
- Autowiring

2. SPRING BOOT (Setup & Automation)

- Spring Boot Fundamentals
- @SpringBootApplication
- Auto-Configuration (very important for interview)
- Starters & Dependency Management
- application.properties / application.yml

3. SPRING DATA JPA (Database Layer)

- JPA Basics (ORM concept)
- @Entity, @Id, @GeneratedValue
- Repository (JpaRepository)
- CRUD Operations
- Query Methods
- @Query Annotation
- Pagination & Sorting (basic idea)

4. SPRING MVC + REST API (WITH DATABASE)

- REST Principles @RestController & Controllers
- Request Mapping: @GetMapping, @PostMapping, @PutMapping, @DeleteMapping
- Request Handling: @RequestBody, @PathVariable, @RequestParam
- DTOs (Data Transfer Objects)
- Validation: @Valid, basic annotations
- Exception Handling @ControllerAdvice, @ExceptionHandler
- MOST IMPORTANT SECTION Goal: Build complete REST API with DB

5. SPRING SECURITY (BASIC + JWT)

- Spring Security Basics
- Authentication vs Authorization
- JWT (JSON Web Token) (Role-Based Access Control (RBAC)
- Secure APIs (Interview + Real-world)

Course Project

Project: Secure User Management REST API

A backend system to authenticate users and manage user data securely using REST APIs and JWT.

Features

- User Registration & Login
- JWT-based Authentication
- Role-Based Access Control (RBAC)
- CRUD operations on users

6. MICROSERVICES

- Microservices Architecture Basics
- Monolith vs Microservices
- API Gateway
- Service Discovery (Eureka)
- Load Balancing (concept only)
- Feign Client (Service-to-Service call)

7. SPRING AI

- Introduction to AI in Backend Applications
- What is Spring AI
- Integrating AI APIs (like OpenAI)
- Basic Prompt Handling
- Simple Program by using SpringBoot with AI

Integration and Deployment

- Configuring CORS for Production and Development Environments
- Integrating React API Client with Authentication
- Implementing RBAC in the Frontend
- Finalizing the Deployment Strategy

Docker Deployment & CI/CD with Jenkins on AWS

- Introduction to Docker
- Dockerfile Creation
- Docker Compose for Multi-Container Applications
- Running a SpringBoot App with Docker
- Automating Deployment with Makefile
- Creating and Pushing Docker Images to Docker Hub
- Introduction to Jenkins and CI/CD
- Setting Up Jenkins for Docker Builds
- Automating Docker Builds and Deployments with Jenkins
- Deploying Docker on AWS EC2
- AWS EC2 Security and Networking

Tools

- Git
- Docker ****
- Makefile
- Jenkins
- AWS