

# Full Stack Web Development Java

**Java, Spring, Spring Boot,  
Hibernate. React.js**

**Duration: 90 Days**



**Register Here**

 **7770043825**

 **406'White Square,  
Hinjewadi, Pune.**



**spring<sup>®</sup>**



**spring boot**



**React**



+91 7770043825



[www.wisdomsprouts.in](http://www.wisdomsprouts.in)



[trainings@wisdomsprouts.in](mailto:trainings@wisdomsprouts.in)

# Syllabus Summary

- HTML
- CSS
- Bootstrap
- Javascript
- React.js
- Core Java
- Advanced Java
- Spring
- Spring Boot
- Hibernate
- MySQL
- Overview of AWS
- Git & Github
- Basic Website Optimization
- Bonus Content
- Sample Mini Project
- Sample Mega Project



## HTML

- Introduction to HTML
- Tags, Elements, and Attributes
- Basics of Syntax
- Creating Tables
- Working with Lists
- Building Forms
- Structure of HTML5
- Semantic and Non-semantic Tags
- HTML5 Features
- New Input Types
- Form Attributes
- Introduction to SVG (Scalable Vector Graphics)
- Canvas Element for Graphics
- Implementing Audio and Video Tags

## CSS

- Using Attributes (ID, Class, Style, Title)
- CSS Types (Inline, Internal, External)
- Understanding the Box Model
- Display Property (Block, Inline, None)
- Visibility and the Hidden Attribute
- Positioning Elements (Static, Relative, Absolute, Fixed)
- Z-index Property for Stacking Order
- Working with Combinators (Descendant, Child, Adjacent Sibling, General Sibling)
- Exploring CSS Pseudo-classes (Link, Visited, Hover, Active)
- Utilizing CSS Pseudo-elements (First Line, First Letter, Before, After, Selection)
- Creating a Static Web Page
- Implementing the Viewport Meta Tag for Responsiveness

## CSS3

- Background, Multiple Backgrounds
- Font Related Features (online fonts)
- Text-Effect and Box-Effect
- Gradients-Linear and Radial
- Transition
- Transformation
- Animation
- Media Queries



## Bootstrap

- Introduction to Bootstrap
- Bootstrap Grid System and Responsive Design
- Bootstrap Components (Navbar, Carousel, Modal, etc.)
- Customizing Bootstrap Themes
- Introduction to Web Performance Optimization
- Minification and Compression of Assets
- Lazy Loading Images and Optimizing CSS/JavaScript

## Javascript

- Introduction to JavaScript
- Variables, Data Types, and Type Conversion
- Operators and Expressions
- Conditional Statements (if, else, switch)
- Looping Statements (for, while, do-while)
- Function Declarations and Expressions
- Scope and Variable Visibility
- Handling Objects and Arrays
- String Operations
- Document Object Model (DOM)
- Selecting and Modifying HTML Elements
- Creating and Removing Elements Dynamically
- Event Handling and Listeners
- Event Handling with Event Delegation
- Dynamic Content Updates without Page Refresh
- Fetching Data from APIs and Updating the DOM

## React.js

- Introduction to React.js and its core concepts
  - Components, JSX, and the Virtual DOM
- Building your first React component
  - Functional components vs. class components.
  - JSX syntax and rendering elements
- Props and State
  - Passing data to components using props
  - Managing component state using setState
- React Router and Single-Page Applications
  - React router
  - Route parameters and dynamic routing
  - Nested routes
  - Protecting routes with authentication guards



- State Management with Redux (or Context API)
  - Introduction to Redux
  - Actions, reducers, and the Redux store
  - Dispatching actions and updating state
  - Async actions with Redux Thunk or Redux Saga
  - Redux project structure
- React Hooks and Functional Components
  - useState, useEffect, and other built-in hooks
  - Fetching data from APIs
  - Custom hooks
- Reusable UI Components
  - Component libraries (e.g., Material-UI, Ant Design)
  - CSS-in-JS, styled-components, and CSS modules
  - Creating a library of reusable UI elements
- Consuming RESTful APIs
  - Axios and Fetch API
  - API responses and errors
  - Integrating API data into React components

## Core Java

- Basics
  - Introduction to Java
  - Setting up Java Development Environment
  - Variables, Data Types, and Operators
  - Control Flow (if-else, switch, loops)
  - String operations
- Object-Oriented Programming (OOP)
  - Classes and Objects
  - Inheritance and Polymorphism
  - Encapsulation and Abstraction
  - Interfaces and Abstract Classes
- Exception Handling and File I/O
  - Exception Handling (try-catch, throws, throw)
  - Checked vs. Unchecked Exceptions
  - File Input/Output (I/O) Operations
- Collections Framework and Generics
  - Arrays vs. Collections
  - List, Set, Map interfaces
  - Generics in Java
  - Iterators and Enhanced for Loop



## Advanced Java

- JDBC (Java Database Connectivity)
  - Introduction to Databases and SQL
  - JDBC Basics (Connection, Statement, ResultSet)
  - CRUD Operations with JDBC
  - Servlets and JSP (JavaServer Pages)
- Introduction to Web Applications
  - Servlets: Lifecycle and Handling HTTP Requests
  - JSP: Syntax, Expression Language (EL), JSTL
  - Building a Simple Web Application
- JavaBeans
  - Understanding JavaBeans
  - Creating and Using JavaBeans
  - JavaBeans in Web Applications
- Advanced JSP and Custom Tags
  - JSP Standard Tag Library (JSTL)
  - Creating Custom Tags (Tag Libraries)
  - Model-View-Controller (MVC) Architecture

## Spring

- Introduction to Spring
  - Spring Framework Overview
  - Dependency Injection and Inversion of Control (IoC)
  - Spring Configuration (XML and Annotations)
- Spring MVC
  - Spring MVC Architecture
  - Handling Requests and Responses
  - View Resolvers and Templates
- Spring Data Access with JDBC
  - JDBC Template and Data Sources
  - Performing Database Operations with Spring JDBC
- Spring AOP (Aspect-Oriented Programming)
  - Understanding AOP Concepts
  - AspectJ and Creating Aspects
  - AOP Integration with Spring
- Spring Transaction Management
- Spring Security



## Spring Boot

- Introduction to Spring Boot
  - Spring Boot Fundamentals
  - Building a Simple Spring Boot Application
  - Spring Boot Starters and Auto-Configuration
- RESTful Web Services with Spring Boot
  - Creating RESTful Endpoints
  - Consuming RESTful Services
  - Spring Boot and REST Security
- Spring Boot Data Access with JPA
  - Integrating Spring Boot with JPA/Hibernate
  - CRUD Operations with JPA Repositories
- Spring Boot and Microservices
  - Microservices Architecture
  - Building Microservices with Spring Boot
  - Service Discovery and Communication
- Spring Boot Security
- Spring Boot Auto Configuration

## Hibernate

- Introduction to Hibernate
  - Object-Relational Mapping (ORM)
  - Hibernate Configuration and Session Factory
  - Mapping Java Classes to Database Tables
- Hibernate Query Language (HQL)
  - HQL Syntax and Querying
  - Criteria API
  - Native SQL Queries with Hibernate
- Hibernate Associations and Caching
  - Mapping Relationships (One-to-One, One-to-Many, Many-to-Many)
  - Second-Level Caching in Hibernate
  - Optimizing Database Access
- Spring and Hibernate Integration
  - Integrating Spring with Hibernate
  - Transaction Management with Spring and Hibernate
- Transactions Management



## MySQL

- Introduction to databases
- MySQL vs NoSQL
- Data Definition Language and Data Manipulation Language
- Data Types
- SELECT statement for retrieving data
- Filtering data with WHERE clause
- Sorting data using ORDER BY
- Joins and relationships between tables (INNER JOIN, LEFT JOIN, etc.)
- Combining multiple conditions using logical operators
- Grouping and aggregating data with GROUP BY and HAVING
- Subqueries and correlated subqueries
- INSERT, UPDATE, and DELETE statements
- Modifying data in tables
- DISTINCT keyword for eliminating duplicates
- LIMIT and OFFSET clauses for pagination
- Using wildcards for pattern matching (LIKE)
- Updating records with UPDATE statement
- Deleting records with DELETE statement
- Introduction to database indexing
- Creating and managing indexes
- Stored Procedures and Triggers
- Backup, Restore, and Security

## Overview of AWS

- Cloud computing overview
- Introduction to AWS
- AWS Service Categories - Compute, Storage, Database, Networking, Security, etc.
- AWS Management Console
- Regions and Availability Zones
- EC2
- S3
- RDS, DynamoDB
- VPC, Security Groups
- Cloudwatch
- Lambda
- Best practices and cost management





## Git & Github

- Version Control with Git and GitHub
- Introduction to Version Control and Git Concepts
- Cloning, Committing, and Pushing Code to Repositories

## Basic Website Optimization

- Basics of Web Performance Optimization
- Minification and Compression of Assets
- Lazy Loading Images and Optimizing CSS/JavaScript
- Web Accessibility Considerations and Practices

## Bonus Content

- Guest Lectures by Industry Professionals
- Mock Interviews and Resume Building Workshops
- Guidance on Building a Professional Portfolio
- Soft Skills Development: Communication, Teamwork, Problem-Solving
- Mid-Course Assessment to Gauge Understanding and Progress
- Job Interview Preparation and Mock Interview Sessions
- Assistance in Job Placement and Internships
- Access to Alumni Network and Industry Events



## Sample Mini Projects

- Personal portfolio website that showcases the student's skills and projects. The website would include
  - A homepage with a brief introduction and profile picture.
  - Sections for displaying projects, each with a title, description, and images/screenshots.
  - A contact section with a form for visitors to send messages.
  - Utilize HTML, CSS (including Bootstrap), and possibly a bit of JavaScript for form validation
- Interactive image gallery that displays a collection of images and provides user-friendly features. The project would involve:
  - Creating a grid of images using HTML and CSS.
  - Implementing lightbox functionality: clicking on an image opens a larger version with navigation buttons.
  - Adding filters or categories to sort images dynamically using JavaScript/jQuery.
  - Applying CSS for a visually appealing and responsive design.
- Simple to-do list web application that allows users to manage tasks. This project can cover various concepts including:
  - Designing the user interface using HTML, CSS, and Bootstrap.
  - Using JavaScript/jQuery to handle tasks like adding, deleting, and marking tasks as completed.
  - Implementing local storage to persist tasks even after page refresh.
  - Enhancing the user experience with animations and transitions.

## Sample Mega Project

- Web platform where users can search and browse real estate listings. The application will allow real estate agents to list properties for sale or rent, and potential buyers or renters can search for properties based on various criteria
  - User Authentication and Profiles
  - Property Listings
  - Search and Filter Functionality
  - Interactive Map
  - Property Details
  - User Reviews and Ratings
  - Saved Searches and Alerts



## Sample Course Completion Certificate



## Sample Internship Completion Letter



Course Details		
1	Duration	180 days
2	Mode of Training	Offline/Online
3	Registration Fees	₹ 999



**Register Here**

