



JAVA **FULL STACK DEVELOPER PUNE**

"Get Placed At Your Dream Company"



START BUILDING
RICH, FUNCTIONAL APPS &
LAND A SOFTWARE
DEVELOPER JOB

100%
PLACEMENT
RECORDS



INSPIRIA

FULL STACK JAVA DEVELOPER PUNE



Explore Program Insights

01

About Us

02

Our Services

03

Full Stack Roadmap

04

Course Curriculum

05

Learning Resources

06

Industry Projects

07

Certification

08

Career Opportunities

09

Meet our Industry Expert

10

Testimonials

11

Admission Process

12

Developers Life



About Us

Embark on Your Full Stack Coding Journey

Greetings Future Developers,

*Welcome to Full Stack Java Developer, where we offer an unparalleled Full Stack Java Industrial Training Program that will equip you with the skills and knowledge necessary to embark on a successful career in **Full Stack Development**.*

Brief overview

At Full Stack Java Developer, Pune- We strive to become a leading provider of Full Stack Java Development course, known for our innovative institute. We aspire to develop a paradigm shift between the student community that technology is easy-to learn & easy-to implement so as to develop real time solutions to the modern problems. **Our Aim is Simple, We strive to create High-impact , Hands-on experiences that prepare learners for meaningful and productive careers.**

Mission

"Our mission is to provide high-quality Full Stack Java Development Industrial Training through a comprehensive offline as well as online platform."

Vision

"Our vision is to empower individuals with the skills and knowledge required to become proficient Full Stack Java Developer and advance their careers."

Whether you're a Beginner looking to get Started with Programming or an Experienced Developer looking to Upskill, *Our Program has everything you need to Succeed !*



Our Services

Strategic Guidance, Exceptional Results.



Why Choose Us

- ✔ 20+ Industrial Projects
- ✔ Build real-world applications like Amazon, Walmart, Zomato
- ✔ Industrial Training by Industry Expert & Assistance with job Placement and Career Guidance
- ✔ Live Doubt Resolutions
- ✔ Interview Preparation & Mentoring
- ✔ Practical Experience through Real-World Projects
- ✔ Industry-relevant projects to develop a portfolio and gain practical experience
- ✔ 400% Salary Hike with Structured Program & dedicated support
- ✔ Access to an active community network of Full Stack Java Developers
- ✔ Industry relevant learning, designed for Fresh Graduates
- ✔ Guidance on best practices for Code Quality, Security & Scalability
- ✔ Personalized feedback from Experienced Instructor and Industry Experts
- ✔ Industry recognized certificate offered by Full Stack Java Developer
- ✔ Our Institute offers a fully-equipped software lab where you can get hands-on practice with the latest tools and technologies.
- ✔ 100% Placement Records with 5* Rated on Google

Please Scan QR Code for More Details

Contact us for More Information-

Mobile : +91 7887575991/ 9356943970

Phone : 020 47212644



Program Highlights

This comprehensive program is meticulously crafted to equip you with the skills and knowledge needed to thrive in the dynamic world of full-stack development using Full Stack Java technologies.



What You will Learn

You will learn everything necessary to **design , develop, and deploy** your application seamlessly, by mastering Full-Stack Development including both frontend and backend technologies, and integrating essential database tools into your skillset.





FULL STACK JAVA DEVELOPER



5 Advance Tools

GitHub.....
 Docker.....
 Kubernetes.....
 Jenkins.....
 AWS.....
 Postman API.....
 Swagger UI.....

4 Front-End

HTML5.....
 JS.....
 CSS.....
 Bootstrap.....
 React JS.....
 TypeScript.....
 Vue JS.....

2 Frameworks

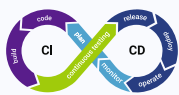
Hibernate.....
 Spring Core.....
 Spring MVC.....
 Data JPA.....
 Junit Mockito.....
 Spring Boot.....
 Spring Security.....
 Microservices.....

3 Database

MySQL.....
 PostgreSQL.....
 SQL Server.....
 MongoDB.....
 Oracle.....

1 Back-End

Core Java.....
 J2EE.....
 JDK 8 | 11 | 17 Features.....



Maven Gradle

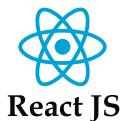




Unleash Your Potential Master The Latest Tech Stack!



Tools and Technologies Covered



**Advanced Software Development Concepts and
Become a 10x Developer using GenAI**



COURSE CURRICULUM

Beyond Books: Your Gateway to a Transformative Learning Experience

Core Java



Introduction To Java

- Java History
- Java Features
- Environment Setup
- Hello World Program
- JDK, JRE, JVM
- JVM Architecture
- Java Comments

Java – Object and Classes

- Class
- Object
- Interface
- Package Creation
- Data types in Java
- Methods
- Variables
- Reserved Keywords
- Coding Standards

Java Basics

- Scanner Class
- Arrays
- Date Concept
- Access Modifiers (private, default, protected, public)
- Static Concept (Block, Class, Method, Variable)
- Final Concept (Class, Method, Variable)
- Type Casting
 - Upcasting
 - Downcasting
 - Autoboxing
 - Unboxing

Java Basic Operators

- Arithmetic Operators
- Assignment Operator
- Relational Operators
- Logical Operators
- Ternary Operators
- Bitwise Operators

Java-Control Statements & Loops

- Conditional Statements
 - if
 - if-else
 - switch case
- Break & Continue Statements in Java
- Control-Flow Statements
 - for
 - while
 - do-while

Object Oriented Concepts

- Abstraction
- Encapsulation
- Keywords- extends, super & this
- Constructor- Default & Parameterized
- Method Overloading & Overriding
- Abstract Class vs Interface
- Aggregation & Composition
- Polymorphism
- Inheritance

Exception Handling

- Exception Basics
- Java Exception Hierarchy
- Exceptions Types in Java
 - Checked Exceptions
 - Unchecked Exceptions
- try, catch, throw, throws, finally
- try with resources
- Create custom/User-Defined Exception

String Concepts

- String Basics
- StringBuffer
- StringBuilder
- String Constant Pool vs Heap
- Repeated Characters with Occurrences
- Repeated Words with Occurrences
- Reverse String without inbuilt function
- String Methods
- String Comparison
- String Concat

File Handling

- File Handling Basics
- Create Folder
- Create File- .txt, .pdf, .xlsx, .docs, .java, etc.
- Write File
- Read File
- Delete File
- Delete Folder

Collection Framework

- List
 - ArrayList
 - Vector
 - LinkedList
- Set
 - HashSet
 - TreeSet
- Map
 - HashMap
 - TreeMap
 - Hashtable
- Collections class
- Differentiate Comparable & Comparator
- LinkedHashSet
- LinkedHashMap
- ConcurrentHashMap
- Fail-Fast Iterator
- Fail-Safe Iterator
- Internal Working of HashMap
- Advanced Collection with Stream API
- Advanced Collection with Java 8 Features

Thread

- Thread Basics
- Thread Life Cycle
- Extends Thread
- Implements Runnable Interface
- Multithreading
- Thread Synchronization
- Thread Methods
- Thread Priorities
- Inter-Thread Communication

JDK 8 Features

- Lambda Expression
- Functional Interface
- Default Method in Interface
- Static Method in Interface
- Method References
- Date Time API
- Stream API in Detail [All Advanced concepts as per Industry Practices]
- Collectors
- forEach Loop
- String Joiner Class
- Parallel Sort
- Optional Class

JDK 11 Features

- Running Java File with single Command
- New Utility Methods in String Class
- Local-Variable Syntax for Lambda Parameters
- Nested Based Access Controls
- JEP 321 : HTTP Client
- Reading/Writing Strings to & From the Files
- JEP 328 : Flight Recorder

Core Java Advance Concepts

- Marker Interface
- Garbage Collection
- transient keyword
- volatile keyword
- covariant return Type
- final, finally & finalize()
- Create Custom Immutable Class
- SOLID Principles
- Enums in Java
- equals() & hashCode() contract
- map() vs flatMap()
- Java instanceOf
- Singleton Class in Java
- SDLC(Software Development Life Cycle) & it's Phases

Projects

- Bank Application – Core Java
 - Sign Up
 - Sign In
 - Deposit
 - Check Balance
 - Withdraw
 - Transfer
 - OTP Functionality
 - LogOut
- E-mail Application – Core Java
- GST Billing Application – Core Java



1. JDBC

- JDBC Introduction & Overview
- MySQL Database Benefits
- Create Database, Table
- SQL Queries
- Joins in SQL
 - Inner Join
 - Left Outer Join
 - Right Outer Join
 - Full Outer Join
- Aggregate Functions
- Java Application Using JDBC Connectivity
- Store and Read Image in SQL
- Handling SQL Exceptions
- View and Indexing
- Stored Procedure

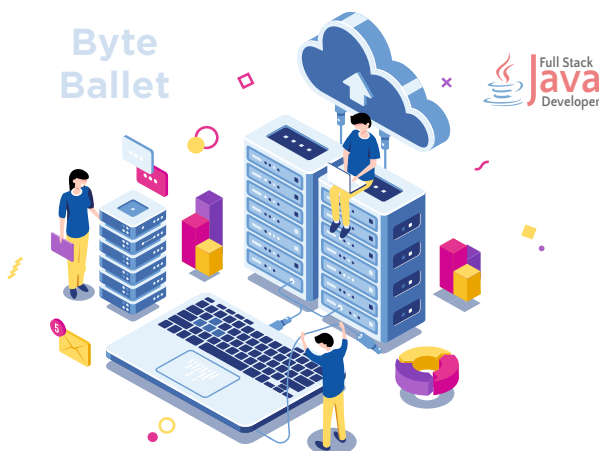
■ Banking Application using JDBC, MySQL Database

2. Servlet/JSP

- Servlet, JSP Basics
- Servlet Hello World Application
- Servlet, JSP Integration
- Deployment Descriptor/Web.xml Structure
- SignUp, SignIn, Read, Update, Delete Operations using, JDBC, Servlet & JSP

Projects

HRM Full Stack Application- JDK 8, Servlet JSP, MySQL, JDBC & Apache Tomcat



Frameworks



Hibernate

- Hibernate Introduction & Overview
- Hibernate Architecture
- ORM Overview
- Hibernate Caching
- Hibernate Session & SessionFactory
- Hibernate Configuration
- Hibernate CRUD Application
- Hibernate Association Mapping
 - One to One Mapping
 - One to Many Mapping
 - Many to One Mapping
 - Many to Many Mapping
- Hibernate Named Queries
- HQL- Hibernate Query Language
- Hibernate HCQL
- Diff. save(), saveOrUpdate(), update(), persist() and merge()
- Diff. get() and load()
- Hibernate Full Stack CRUD Application

Projects

HRM Full Stack Application- JDK 8, Servlet JSP, MySQL, Hibernate & Apache Tomcat



Spring Framework

- Spring Framework Introduction
- Spring Framework Features
- Inversion of Control Spring (IoC)
- Dependency Injection
- Spring Beans LifeCycle
- Beans Scope
 - Singleton
 - Prototype
 - Request
 - Session
 - Global-Session
- Auto wiring
- Spring Framework Annotations



Spring Boot

- Spring Boot Introduction
- Spring Boot Features
- Spring Boot Hello World Application
- Spring Boot Annotations
- Spring Boot JAR File Creation
- Spring Boot WAR File Creation
- Spring Boot CLI
- Spring Boot Actuator
- Spring Boot with Logger
- Deploy Spring Boot Applications on External Tomcat Server
- Spring Boot – application.properties
- Spring Boot with Hibernate Integration
- Spring Boot with Hibernate CRUD
- Spring Boot with Spring Data JPA Integration
- Spring Boot with Spring Data JPA CRUD App
- Spring Boot with JDBCTemplate Integration
- Spring Boot with XML Payload
- Spring Boot with JSON Payload
- Spring Boot–Change Default Tomcat Server Port
- Spring Boot with Docker Implementation
- Spring Boot with Junit Testing
- Spring Boot with All HTTP Methods
 - GET ■ POST ■ PUT ■ DELETE ■ PATCH
- Spring Boot with HTTP Status Code
- Spring Boot with GitHub
 - Git Clone ■ Git Add ■ Git Commit
 - Git Push ■ Git Merge ■ Git Pull
 - Git Branch ■ Git Stash ■ Git Checkout
- Spring Boot with Swagger UI Configuration
- Spring Boot Global Custom Validations
- Spring Boot Global Custom Exception
- Spring Boot Code Debug Practices in Eclipse and IntelliJ Idea
- Spring Boot Application connectivity with Multiple Databases
- Spring Boot Realtime Excel File Upload
- Spring Boot Realtime Image Upload
- Spring Boot Custom Annotation
- Spring Boot with Caching Mechanism
- Spring Boot with Thymeleaf
- Spring Boot with GraphQL
- Spring Boot Data JPA Entity Mapping
 - One to One ■ One to Many
 - Many to One ■ Many to Many
- Spring Boot with Spring Batch
- Spring Boot with Reactive Programming
- Spring Boot with Splunk Integration
- Spring Boot External API Call–
 - RestTemplate ■ FeignClient ■ WebClient
- Spring Boot with JDK 8 Features
- Spring Boot with JDK 11 Features
- Spring Boot 3.x with JDK 17 Features
- Spring Boot 2.x to 3.x Migration using Rewrite
- Spring Boot with OAuth Implementation
- Spring Boot with JWT Implementation
- Spring Boot Java Email API Implementation
- Spring Boot send SMS/OTP
- Spring Boot 3+ with Swagger UI
- Payment Gateway Live Implementation
- Spring Boot with Advanced JPAQL
- Spring Boot with JUnit Mockito
- Spring Boot with OpenAI ChatGPT Integration
- Spring Boot Application Live Deployment
 - AWS Cloud ■ GCP Cloud
- Spring Boot Code Coverage
 - SonarQube ■ Jacoco
- Spring Boot with Database
 - MySQL ■ PostgreSQL
 - MongoDB ■ Oracle
- Spring Boot with React JS
- Spring Boot with TypeScript
- Spring Boot with Vue JS
- CI-CD Pipeline Using Jenkins
- Spring Boot API Documentation
- **Spring Boot Advanced Research & Development**

Projects

- Spring Boot Chat Application
- Spring Boot QR Code Application
- Spring Boot HRM Application
- Spring Boot GST Billing Application
- Spring Boot E-commerce Application
- Spring Boot Finance Banking Application

Spring Data-JPA

- Spring Data JPA Basics
- Spring Data JPA vs Hibernate
- Spring Data JPA Architecture
- Spring Data JPA application.properties file
- JPA Repository vs CRUD Repository
- Spring Boot with JPA Caching Mechanism in Service Layer
- Data JPA Custom methods and JPA Query Language(JPAQL)
- Spring Boot with Spring Data JPA CRUD
- Spring Boot with Spring Data JPA
 - Custom Validation ■ Custom Exception
- Spring Boot Paging & Sorting with JPA
- Data JPA Transaction Management
- **Spring Boot HRM Application with Data JPA using Swagger UI**

Spring Security

- Spring Security Basics
- Spring Security Features
- Spring Security Authentication
- Password Encryption by using Spring Security
- Spring Security Login Form
- Spring Security OAuth2 RealTime Implementation
- Spring Boot End Points Security
- Spring Boot, Spring Security- Restful Web Services with Advanced Authentication
- Single Sign On with Spring Boot & Okta | OAuth 2.0
- Spring Boot with JWT Token base authentication
- Spring Boot with Spring Security using BCrypt Algorithm Role base authorization
- Spring Security CRUD- SignUp & SignIn Application
- **Spring Boot, Spring Security, JWT GST Billing Application**
- **Spring Security Advanced Research and Development**

Spring- JDBC Template

- JDBCTemplate Introduction
- Builder Annotation
- Spring Boot with JDBCTemplate using Swagger UI
- Spring Boot with JDBCTemplate CRUD Application
- Spring Boot with JDBCTemplate SignUp | SignIn Application
- Spring Boot with JDBCTemplate Java 8 Features Live Implementation
- **Spring Boot HRM Application with JDBC Template Advanced Swagger UI**

Spring MVC

- Spring MVC Basics
- Spring MVC Architecture
- Spring MVC Hello World Application
- Spring MVC with Maven
- WAR File Deployment on External Tomcat
- Spring MVC HRM Full Stack Application

Projects

CRM Application using JDK 8, Spring MVC, Spring Security, Hibernate, Maven, MySQL & Apache Tomcat

■ Modules

- Customer SignUp, SignIn
- Add, View , Edit , Delete
- Add Roles- Admin, DBA, User
- Password Encryption
- Remember Me | Timeout Functionality
- Deploy Application on External Tomcat Server



JUnit- Mockito Framework 5

- JUnit Introduction & Overview
- Mockito Introduction
- Mockito Annotations
- Mockito Setup
- What is Mocking?
- Need for Mocking
- Methods of Mockito
- Mockito mock() method
- Mockito when() method
- Mockito verify() method
- Mockito doNothing() method
- Mockito times() method
- PowerMock
- JUnit Mockito Hibernate Integration
- JUnit Mockito JPA Integration
- JUnit Mockito JDBCTemplate Integration
- JUnit Mockito Java 8 Features Live Implementation
- **HRM Application using JUnit Mockito**
- **GST Billing App using JUnit Mockito**
- **Ecommerce Application using JUnit Mockito**

MongoDB



- NoSQL DB Introduction
- MongoDB Features
- MongoDB Database & Collection
- Spring Boot with MongoDB Integration
- Spring Boot with MongoDB CRUD
- Spring Boot API Creation with MongoDB
- Spring Boot with MongoDB using Java 8 Features Live Implementation
- Spring Boot with MongoDB Full Stack
 - React JS
 - TypeScript
 - Vue JS

Projects

- Spring Boot Full Stack HRM Application with MongoDB
- Spring Boot Ecommerce Application with MongoDB

Web Services

SOAP Web Service

RESTful Web Service

Design Patterns

Creational design Patterns

- Singleton Pattern
- Factory Pattern
- Abstract Factory Pattern
- Prototype Pattern
- Builder Pattern
- Object Pool Pattern

Structural design Patterns

- Adapter Pattern
- Bridge Pattern
- Composite Pattern
- Decorator Pattern
- Facade Pattern
- Flyweight Pattern
- Proxy Pattern

Behavioral Design Patterns

- Iterator Pattern
- Interpreter Pattern
- Mediator Pattern
- Memento Pattern
- Observer Pattern
- State Pattern
- Strategy Pattern
- Template Pattern
- Visitor Pattern

Microservices Design Patterns

- API Gateway
- Circuit Breaker
- Service Registry
- CQRS Pattern
- SAGA Design Pattern
- Event-Driven Architecture
- Polyglot Persistence

Introduction to AWS (Amazon Web Services)

- AWS Cloud Features
- Amazon Cloud Services-
 - Ec2 ■ Elastic Beanstalk ■ RDS ■ Lambda
- Create EC2 Instance in AWS
- Create Role and User

Deployment

- Deploy and Scale Web Application
- Deploy Spring Boot Application on AWS Elastic Beanstalk
- Upload and Deploy web application in a simplified fast way

Projects

- Spring Boot AWS Live Deployment-
 - GST Billing ● HRM Application
 - Ecommerce ● Banking Application
- AWS Interview Questions & Answers as per Industrial Practices
- AWS Advanced Research and Development - Live Deployment

Microservices



Microservices Intro & Overview

- Monolithic vs Microservices
- Spring Boot with Data JPA Microservices
- Creating Multiple Services
- Logger with Microservices
- Microservices Communication
 - RestTemplate ■ FeignClient ■ WebClient
- VO package creation for Multi Model App
- Set port number and application name for each service
- Service Registry- Eureka Server
- API Gateway, traverse centralized request
- Hystrix DashBoard
- Zipkin & Sleuth
- Trace Distributed Logs
- Spring cloud API Gateway
- Microservices Fault tolerance

Microservices Architecture

- Microservices Project Structure
- Design Standards
- Spring Boot Coding Standards with Microservices
- application.properties vs application.yml file
- Synchronous vs Asynchronous Calls

Microservices Advanced Concept

- Fallback Controller Implementation
- Fault Tolerance in Microservices
- Trace Dependent Services
- Trace Controller | Methods and Services how many times used/hit
- Microservices Load Balancing
- Microservice Application Deployment using Docker
- Spring Boot Project with Microservice architecture Real time Integration of Spring Cloud Eureka Server, Spring Cloud API Gateway, Hystrix, Zipkin & Sleuth
- Microservices Integration with Front End
 - React JS ■ TypeScript ■ Vue JS
- **Microservices Advanced Research and Development**

Projects

- Microservices HRM Application
- Microservices Customer | Product | Payment Order Application
- Microservices Finance Bank Application
- Microservices Ecommerce Full Stack App
 - MySQL ● PostgreSQL ● MongoDB
- Microservices GST Billing Application

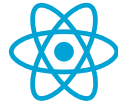


Front End- UI



- HTML 5
- JavaScript
- CSS
- Bootstrap
- Website Deployment and Go Live
- Domain and Hosting
- Payment Gateway Integration

React JS



Introduction to React JS

- What is React?
- Why React JS?
- React JS Application Setup
- Structure of React JS Projects
- React JS Hello World
- React JS JSX
- ES6 Features

React JS Components

- Intro to React Components
- Component lifecycle
- Creating a React Component
 - Class Component
 - Functional Component
- Rendering multiple components
- How to split an App into components
- Comparing Stateless and Stateful Components
- Understanding Higher Order Components

Props, States & Events

- Introduction to Props
- Props in action- adding props to `ReactDOM.render()`
- What is State in React
- Introduction to Events
- Events in action
- Difference between props and State
- Adding state to the component
- Multiple child components
- **Creating App using states and props**

Working with React-Router

- Introduction to React Router
- Server vs Client Routing
- Setting up a 404
- Linking Between Routes

Handling Forms in React App

- React JS form
- Setting Up a JS Configuration for the Form
 - Handling User Input
 - Handling Form Submission
- Adding Custom Form Validation
- Showing Error Messages

React Redux

- Introduction to Redux
- Understanding the Redux Flow
- What is an Action ?
- What is Store ?
- Connecting React to Redux
- Connecting the Store to React

React Hooks

- What is a Hook?
- React useState Hook
- React useEffect Hooks
- React useContext Hook
- React useRef Hook
- React useReducer Hook
- React useCallback Hook
- React Custom Hooks
- **React Advanced Research & Development**

React Js Back-End Spring Boot API Integration

- Axios
- Fetch

Projects

- QR Code Application
- ToDo App using React JS & Spring Boot
- GST Billing
- HRM Application
- ECommerce Application
- Banking Application

Introduction to TypeScript

- TypeScript Overview
- JavaScript vs TypeScript
- Advantages of using TypeScript
- TypeScript Environment Setup
- Hello World Application in TypeScript
- Writing your first TypeScript function
- TypeScript Variables

Basics Type Constructs

- String and Number Types
- No Emit On TypeScript Error
- String Literal Types
- Union Types and Type Aliases
- Function Parameter and Return Types
- Void and Never Types
- Null and Undefined Types
- Describing values using Interfaces
- Enumerations in TypeScript

Functions in TypeScript

- Understanding functions
- Rest Parameters in functions
- Specifying function parameter types
- Adding function overloads

Classes

- Class Properties
- Static vs Instance Properties
- Class Methods
- Extends and Implements
- Access Modifiers

Generics and Files

- Introducing generics
- Creating generic classes
- Applying generic constraints
- Introduction to ".d.ts" Files
- Definitely Typed and @types
- Creating Custom utility types

Decorators

- Implementing method decorators
- Implementing class decorators
- Implementing property decorators
- Implementing decorator factories

Advanced Type Constructs

- Typed this
- Type Guards
- Type Assertions
- Non-null Assertion Operator
- Intersection Types
- key of and Lookup Types
- Mapped Types

Declaration Merging

- Merging interfaces
- Module augmentation

TS Real World Applications

- Integrating with Libraries/Frameworks
- Promises and async/await in TypeScript
- Asynchronous Operation with TypeScript
- Error Handling Strategies in TypeScript
- Convert JavaScript Project into TypeScript
- TypeScript with Back End API Calling
- TypeScript with Spring Boot Integration
- TypeScript CRUD using Spring Boot
- TypeScript Interview QnA
- **TypeScript Advanced Research and Development**

Projects

- GST Billing
- HRM Application
- ECommerce Application





Introduction to Vue.js

- What is Vue.js?
- Why Vue.js?
- Hello World with Vue.js

Working with Template & ViewModel

- Vue.js template
- viewmodel / vue instance
- Interpolation
- Vue.Js Directive
 - v-bind ■ v-if ■ v-show
 - v-for ■ v-on ■ v-model
- Filters
- Computed properties
- Watchers
- Conditional Rendering
- Iterative Rendering

Vue Instance Revised

- Basic vue instance usage
- Multiple vue instances
- Vue instance Lifecycle
- Accessing data from outside vue instance

Event Handling

- Introduction
- Listening to events
- Method Event handlers
- Event modifiers
- Key modifiers

Working with forms & two way binding

- Introduction to two way binding via
- v-model
- Handling user input with HTML form controls
- Data binding
- .lazy, .number, .trim modifiers
- Submitting a form

Introduction to Components

- What are components?
- Why components ?
- Registering Components Locally and Globally
- Composing application as components

Passing data to components

- Props, parent to child communication
- Custom events, child to parent
- Non parent child communication event
- emitter style using .emit, .on

Content distribution with slots

- Compilation scope
- Named slot
- Single slot
- Scoped slot

Dynamic Components

- Switching Multiple Components with
- Dynamic Components
- Dynamic Component Behavior
- Keeping Dynamic Components Alive
- Lifecycle hooks

Custom Directives

- Introduction
- Directive hook functions
- Developing custom directive
- Passing values and arguments to custom directives
- Directive hook arguments

Mixins

- Introduction
- Creating and using mixins
- Option merging
- Global mixin

Vue.Js Hooks(Composition API)

- Understanding Vue.js Composition API
- Lifecycle Hooks
- Custom Hooks & Composition Functions
- Accessing DOMs and External Libraries
- Applying Composition API to a Project

Remote Communication with vue Js

- Using vue-resource to working with HTTP
- Fetch data
- Post data to server
- Request Response Interception
- Template URLs

Routing

- Introduction
- Loading routes
- Routing modes
- Navigation with router links
- Imperative navigation
- Named routes
- Query parameters
- Named router views
- Wildcard routes
- Route guards

State Management

- Introduction
- What is vuex ?
- Centralized store Using Getters
- Mapping Getters to Properties

Understanding Mutations

- Using Mutations
- Using Actions
- Mapping Actions to Methods

Vue.Js Pinia

- Pinia Introduction
- Define your first Pinia Store
- Pinia with the Options API
- State ➤ Getters ➤ Actions
- Pinia Vue.Js State Management
- Advanced Pinia
- Spring Boot with Vue JS Integration
- Vue JS CRUD using Spring Boot
- **Vue JS Advanced Research & Development**







































Projects

- GST Billing
- HRM Application
- ECommerce Application
- Banking Application



Advance Industrial Tools & Technologies

-: Industry Tools :-

 Maven	 Apache Kafka
 Gradle	 Oauth
 Project Lombok	 GitHub
 JWT	 Jenkins
 JUnit	 JIRA
 Mockito	 Ansible
 Jacoco	 Docker
 SonarQube	 Kubernetes
 Postman API	 DevOps
 Swagger UI	 Splunk
 Thunder Client	 AWS Cloud
 JSON	 GCP Cloud
 Java Mail API	 SOLID Principles
 JAR & WAR	 GitLab
 Apache Tomcat	 BitBucket
 GraphQL	 Material UI
 Netflix Eureka	 Next JS
 Hystrix	 Node JS
 Zipkin	 Axios
 Redis	 JEST
 Redux	 ChatGPT
 Eslint	 Agile

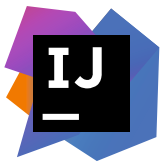
● Industry Softwares ●



JDK 17



Eclipse



IntelliJ Idea



Visual Studio
Code



Apache
tomcat

Maven

Apache
Maven



Gradle



MySQL



SQLYog

ORACLE

Oracle



Mongodb Compass

Compass



Github



SQLWorkbench



mongoDB

MongoDB



POSTMAN

Postman API



Node. Js



Docker

Docker



GitBash

Interview Preparation

- Programming Concepts
- Programming Standards & Practices
- Hands-on Projects
- System Design & Integration
- Software Architecture
- Career Guidance
- Professional Consultation
- Developer Life

- Full Stack Java Developer training in Pune, India Offers both online and offline mode.
- Ideal for fresh graduates, students, developers, and professionals.
- Our program covers a wide range of Full Stack Development topics.

Duration

9

Months



Average Package

INR 7 to 45 LPA

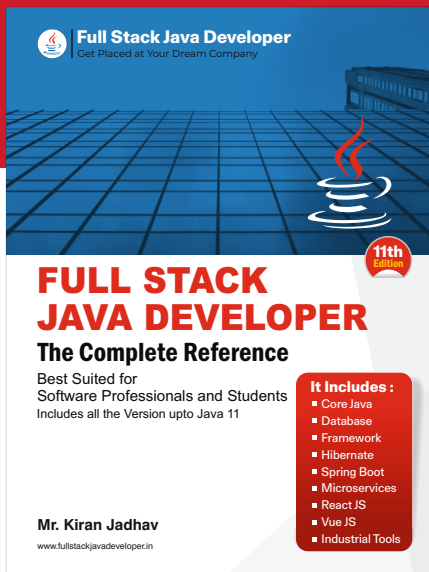
Learning Resources

Expand Your Knowledge with Our Recommended Books !



Full Stack Java Developer Interview Q & A

This book is a comprehensive guide to Full Stack Interview Questions & Answers, covering Core Java, JDK 8, Programs, Databases, Hibernate, Spring Boot, Spring Security, Data JPA, Microservices, React JS, and essential Tools. It includes 201 Full Stack questions from Fortune 500 MNC interviews, providing detailed explanations & professional terminology to help readers excel in their interviews.



Full Stack Java Development Reference Book

Fully updated for Java SE 11, Full Stack : The Complete Reference, Eleventh Edition explains how to develop, compile, debug, and run Java programs. Best-selling programming author Mr. Kiran Jadhav covers the entire Java language, including its syntax, keywords, and fundamental programming principles. Best of all, the book is written in the clear, crisp, uncompromising style that has made the choice of millions worldwide.

Key Features

- Comprehensive guide for Java developers.
- Expert insights and real-world scenarios.
- Perfect for interviews and skill enhancement.
- In depth Coverage
- Interview-Centric Approach

Grab your copy now and Elevate your Full Stack Java Skills!

For more details please visit us at- <http://www.fullstackjavadeveloper.in/>

Full Stack Industrial Projects

Our Industry Projects Elevate Your Practical Expertise !

Projects using Core Java



Core Banking



Billing Application



Java Mail API

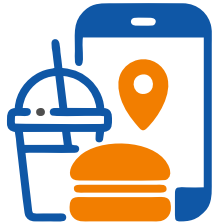


Core HRM

Projects using Spring Boot with React JS



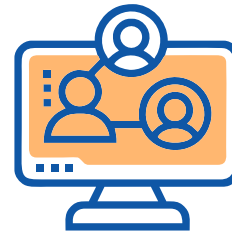
Ecommerce



Food delivery



HealthCare



HRM Application



GST Billing

Projects using Spring Boot with Vue JS



Chat Application



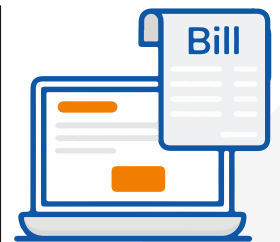
HRM Application



Airlines



Retail Ecommerce



Billing System

Projects using Spring Boot with Microservices & ReactJS



GST Billing



Ecommerce



HRM Application



HealthCare

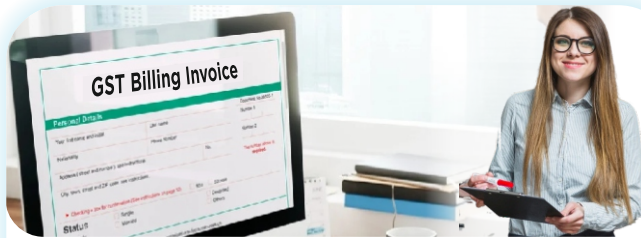


Finance App

Full Stack Java Major Projects

9 Months Training + 3 Months Internship = Success!

Training for Excellence, Internship for Experience



GST Billing

Simplify your billing process with our GST-compliant invoicing solution. Automate tax calculations, generate professional invoices, and ensure regulatory compliance seamlessly.



Ecommerce

Launch your online store effortlessly with our comprehensive eCommerce solution. From product listings to secure checkout, streamline your sales process and reach customers worldwide.



HRM Application

Streamline HR processes with our comprehensive HRM application, from recruitment to payroll. Simplify workforce management (CRUD), boost efficiency, & ensure compliance.



HealthCare

Create an integrated healthcare application using Java Technologies & incorporate the best data security practices applicable to the healthcare industry.

**After 9 months of rigorous training, dive into
3 months of real-world experience during your internship**

Your Full-Stack Development Certificate Awaits



CERTIFICATE
OF ACHIEVEMENT

Full Stack Java Developer Pune



THIS IS TO CERTIFY THAT

MEGHA ROY

Has Successfully completed the Full Stack Java Development
Project-Based Industry Training Program conducted by
Full Stack Java Developer Pune.

Date: 27 / 12 / 2019



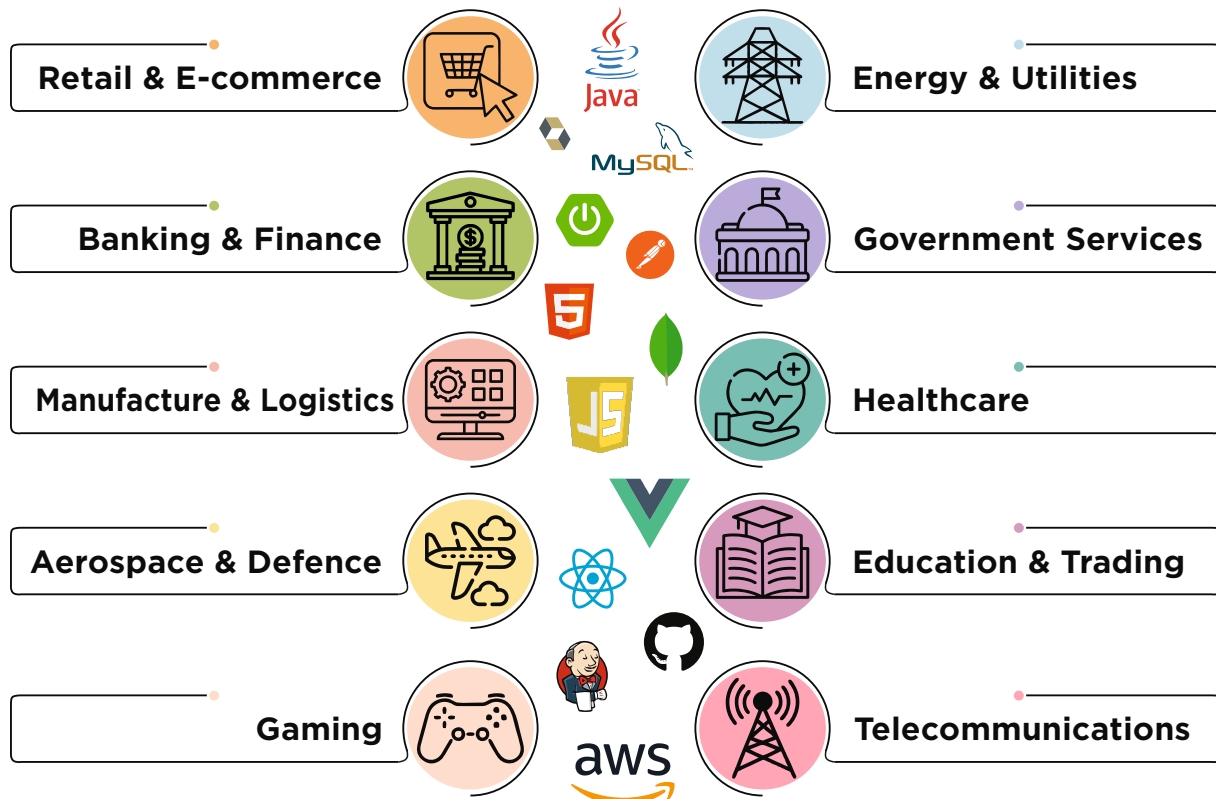
Scan to Validate

K. Jadhav
Kiran Jadhav
Technical Director

Career Opportunities

“We Guide You Towards Your Dream Salary Package”

Full Stack Developers are in High-Demand across World-Wide Industries



Job Roles you can Apply for

Full Stack Developer	Backend Developer	Java Developer
Software Engineer	Associate Developer	Web Developer
Software Developer	Product Developer	Sr Full Stack Developer
Full Stack Team lead	Technical Architect	CTO

Our Alumni work at Reputed Tech Organizations & Promising Startups

Our alumni have successfully transitioned their Careers to Business and Functional roles in Software Industry, working with some of the leading Organisations in the World Including :-



MEET OUR INDUSTRY LEADING EXPERT



Mr. Kiran Jadhav

Software Architect, Full Stack Java Developer Book Author

Bridging Borders through Code

"Whether you're based in Pune, USA or anywhere in between, our course transcends boundaries to deliver a world-class education."

Our Expert Says -

"Work very hard! Those who work hard for the first 3 years of their life enjoy the next 50 years of relaxation and those who spend the first 3 years in relaxation spend remaining years working hard."

Mr. Kiran Jadhav is our industry leading Expert, responsible for designing & delivering a Full Stack Java Development Industrial training program that improve the skills and knowledge of our Students. He holds the B.E. degree in Information Technology from Pune University, First Class with Distinction. He has about 10+ years of industry experience in Developing, Architecting, and Deploying Enterprise-Level Full Stack technological product and solutions for Fortune 500 MNC's in the domain of top Fintech, Retail Ecommerce, Banking, Insurance and Education technology. He brings a wealth of experience in training candidates not only in Pune, India but across Worldwide.

Mr. Kiran Jadhav, author of the two Best Sellers Book Full Stack Java Developer - 'The Complete Reference' and 'Full Stack Java Developer Interview Questions & Answers' Book. Has worked at various corporates as Software Developer, Team Lead, Solution Architect, Program Manager, Software Architect and Corporate Trainer in the last 10 years. Mr. Kiran Jadhav is teaching and interacting with more than 1000+ new candidates every month, including Software Developers and foreign candidates.

Reach At:  jadhavkiran2011_1

 <https://www.linkedin.com/in/kiran-jadhav-22649b99>



1000+

Corporate Workshop



150000+

Happy Candidates



1000+

Campus Training



10+

Years of Industry Experience

See What Our Learners Have to Say



Vishnu Londhe | TCS

I was looking to make a move in my career. Being a college student. I had no clue on how to apply for job ... That's when I heard about Full Stack Java Developer! I immediately applied and started attending classes and got an offer from TCS in just 9 months.



Megha Roy | Zensar Technologies(USA)

During my final year in college, I enrolled to the Full Stack Java Developer Course, and cool thing is, I landed a job in the same year! It showed I could use what I learned in class for real projects and rock in my development career.



Harshad Ramteke | Cognizant

I landed a role as a Software Engineer immediately after completing the course. The comprehensive curriculum and industry-relevant projects made a significant impact on my skill set. Thank-You Full Stack Java Developer for making my Career!!



Prashant Daye | Amdocs

My Journey transformed from novice developer to seasoned expert ! Here you will get everything you need to become a Full Stack Java Developer, No Second Thoughts needed , Enroll now !



Lakshmi Basu | Tech Mahindra(USA)

I am now thriving as a Java Developer, thanks to the strong foundation and project-based learning approach of this program. Highly recommended for aspiring software Developers.



Sachin Bachkar | John Deere

This course paved the way for me to land my dream job! If I can achieve it, so can you! Join now and turn your aspirations into reality.



Admission Process

Learn Coding Today, Achieve Greatness Tomorrow!

Fill out the Registration Form here



Pre-requisite Degree

Any Graduate
(Completed/ pursuing)



Job- Ready Duration

9 Months



Placement

100% Placement Record

SCAN THE QR

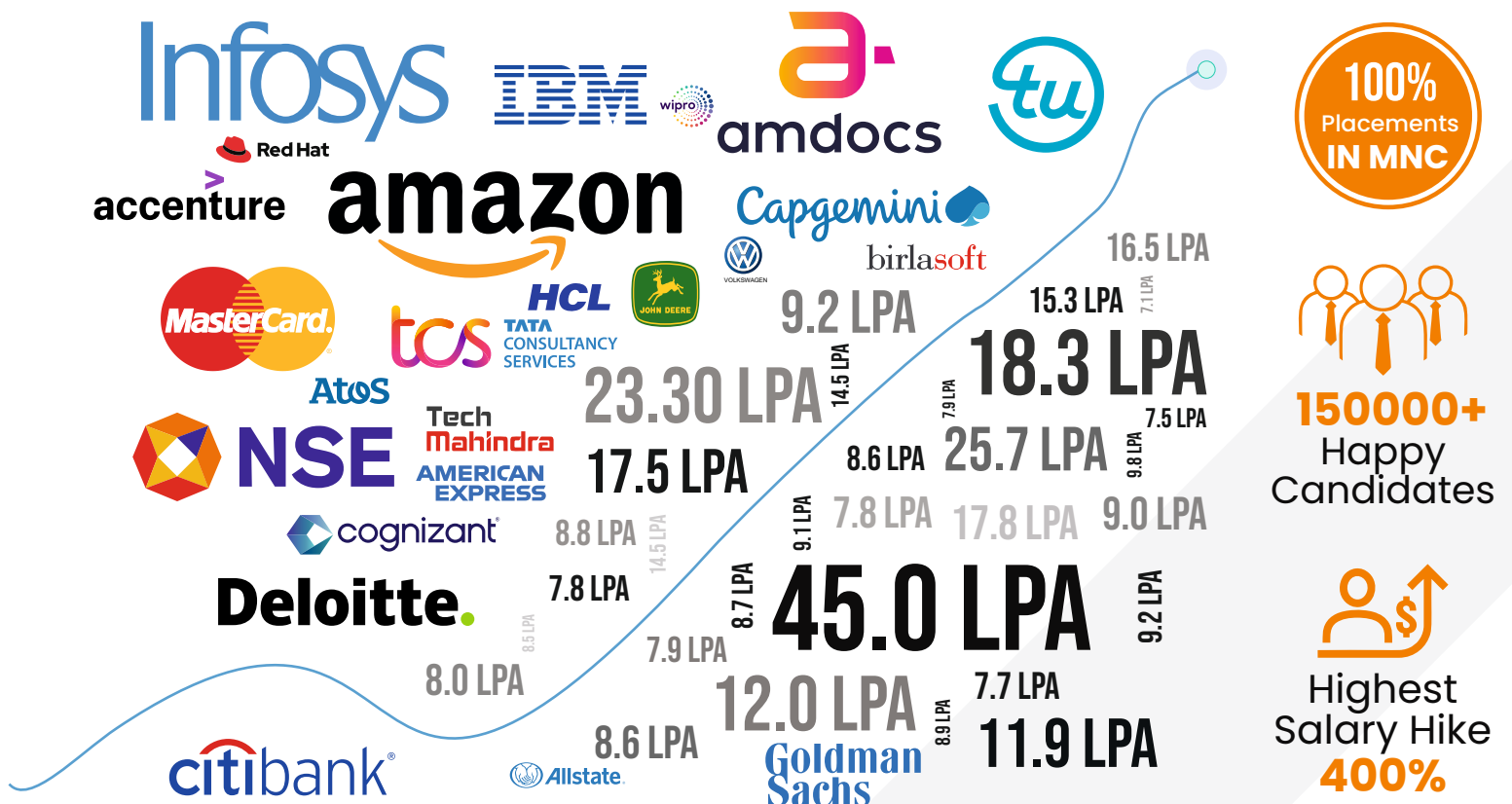
or visit

fullstackjavadeveloper.in

book your Seat Now!



Scan Me





Developer Life

Building Dreams in Bits and Bytes: Life through the Developer's Lens

These Full Stack Developer Life career thoughts are based on the past 10+ years of experience in Software Development Industry. We have gone through different stages in our careers starting from Software Trainee Developer till Solution Architect. We are not going to dictate any of the points, but all the practices listed here contributed a lot in our software development careers, so if you think they make some sense for you then try to adopt it. "

Here are some Coding Best Practices that developers should adopt:

What is Practice?

- Practice is a Habit.
- Practice does not need to remember.
- Practice comes by practicing.
- Practice needs lot of dedication and commitment.

To become a successful software developer, you need lot of practice, dedication and commitment.

Do you read Software Source Code?

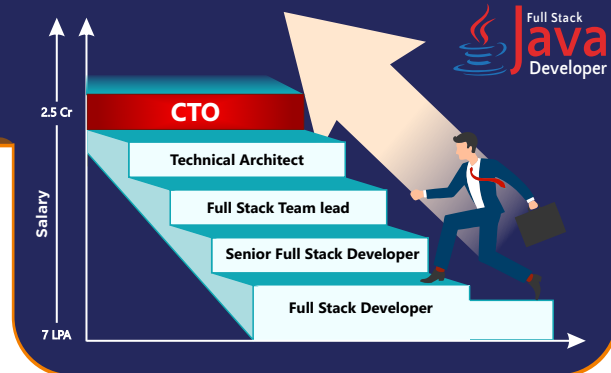
- Only few software developers will have positive answer because reading and understanding an existing software source code is the most boring task. If you are one of them who feels reading software source code is a boring task, then you are missing one of the most important best practices, which a software developer should have in his/her life.
- There are many attributes of software codes (indentation, comments, history header, function structure, etc.), which you will learn by reading existing code, specially, a code written by well- experienced software developers.

Adhere to established standards; avoid inventing new ones.

Most of the standard software organizations maintain their coding standards. These standards would have been set up by well-experienced software developers after spending years with software development. This is equivalent to following footsteps of great people left behind them.

SOLID Principles :

- Single Responsibility Principle (SRP)
- Open-Closed Principle (OCP)
- Liskov Substitution Principle (LSP)
- Interface Segregation Principle (ISP)
- Dependency Inversion Principle (DIP)



Code should be written to be review

While writing your software code, keep in mind that someone is going to review your code and you will have to face criticism about one or more of the following points but not limited to:

- Not following standard.
- History, Indentation, Comments are not appropriate.
- Not keeping performance in mind.
- Too much hard coding.
- Repeated code

Testing to be followed like an Important thing

Testing is mandatory after every small or big change no matter how tight schedule you have or you just changed a small comment inside the code, you have testing due for the changed code.

- Tight schedule, no compromise
- Changed just a comment, still you have to test it
- Changed just a variable name, testing has to be done.
- If you feel lazy it's too dangerous.

Keep your Code & Documents Safe

A smart developer keeps habit of taking daily backup of the produced artifacts, otherwise machine crash can crash you as well. You should keep your artifacts at your local machine as well as another secure machine, so that in case of machine crash, you can continue with the saved copy of the source code or documents.

New technologies are coming everyday

If you want to sustain in the market, then you would have to keep yourself updated with latest IT tools, and technologies. Following are the few sources:

- Technical Forums over the internet.
- Technical magazines on various IT subjects.
- Technical Bulletin Boards
- Conferences, Trainings and Workshops

Career Planning:

Today's professional life is very dynamic and to move along with it we need a proper career planning. When you start your career as a software developer, you really do not know how exactly you will perform in the industry, though you have confidence that whatever you do, will be done in the best way.

We're Here to Help!

Whether you have a question about our course, or just want to share your thoughts with us, We're Here to Guide You!



Learning Center[Research and Development]
**Inspiria Mall, F-16, Level-04, Near Bhakti
Shakti Chowk, Nigdi, Pimpri-Chinchwad,
Old Mumbai-Pune Highway, Pune - 411044**



Get in Touch
contact@fullstackjavadeveloper.in



Give us a call
**Mob.: +91 7887575991/ 9356943970
Ph.: 020 47212644**



www.fullstackjavadeveloper.in

**“Build Tomorrow's Tech with
Full Stack Java Developer, Pune !”**