

★ **Java Full Stack Developer**

■ **Module 1: Introduction to Java Full-Stack Development**

- What is full-stack development?
- Why Java is still the #1 backend language
- Overview: Java → Spring Boot → Database → Frontend
- Client–server architecture
- REST APIs overview
- Development environment setup (JDK, IntelliJ/Eclipse, Postman, Git, VS Code)

■ **Module 2: Core Java (Beginner to Advanced)**

Java Basics

- Variables, Data Types
- Operators
- Conditions and Loops
- Arrays & Strings

Object-Oriented Programming (OOP)

- Classes & Objects
- Inheritance, Polymorphism
- Abstraction & Encapsulation
- Constructors & Method Overloading
- Static & Final keywords

Collections & Generics

- List, Set, Map
- Iterator, Comparable, Comparator
- Generics & Type Safety

Exception Handling

- try/catch, finally
- Custom exceptions

Java I/O

- File handling
- Serialization

Java 8+ Features

- Lambda expressions
- Streams API
- Functional interfaces
- Method references
- Optional API
- Date and Time API

Multithreading & Concurrency

- Threads
- Executor framework
- Synchronization basics

■ **Module 3: SQL & Databases**

RDBMS (MySQL / PostgreSQL)

- CRUD operations
- Joins, Subqueries
- Indexing
- Views & Functions
- Stored Procedures
- Database design & ER diagrams

■ **Module 4: Frontend Development**

✓ **Option A – React.js**

- Components
- Props & State
- Hooks (useState, useEffect)
- React Router
- Handling forms & validation
- API calling using Axios
- Tailwind CSS / Bootstrap
- Building reusable components

✓ **Option B – Angular**

- TypeScript basics
- Components & Modules
- Services & Dependency Injection
- Angular Routing
- Reactive Forms
- HttpClient API calls
- Angular Material UI

■ **Module 5: Introduction to Spring Framework**

- What is Spring? Why it dominates enterprise Java?
- Inversion of Control (IoC) & Dependency Injection (DI)
- Spring Boot vs Spring MVC
- Setting up Spring Boot project (Spring Initializr)

■ **Module 6: Spring Boot 3 – Backend Development**

Core Concepts

- Controllers
- REST APIs
- Request/Response handling
- Path variables & Request parameters
- Exception Handling (Global Exception Handler)
- Validation (Spring Validation API)

Spring Boot Features

- Application properties
- Profiles (dev/test/prod)
- Logging
- Actuator

■ Module 7: JPA & Hibernate (ORM Layer)

- Introduction to Hibernate
- ORM concepts
- Entities & Primary Keys
- One-to-One, One-to-Many, Many-to-Many relationships
- JPQL & Criteria API
- Pagination & Sorting
- Lazy vs Eager loading
- Mapping DTOs

■ Module 8: Spring Data JPA

- CrudRepository
- JpaRepository
- Built-in queries
- Custom queries
- Pagination & sorting
- Projections

■ Module 9: Security & Authentication

Spring Security

- Introduction to authentication & authorization
- JWT authentication (access token + refresh token)
- Password hashing (BCrypt)
- Role-based access control (RBAC)
- Securing REST APIs
- CORS configuration

■ Module 10: Connecting Frontend (React/Angular) with Backend (Spring Boot)

- CORS
- Axios / HttpClient calls
- JWT token flow
- Login/Registration UI
- Protecting routes
- CRUD UI with Backend integration

■ Module 11: File Uploads & Handling Media

- File uploads with Spring Boot
- Storing files in filesystem

- Upload preview in React/Angular
- Serving static and media files

■ Module 12: Testing (Optional)

- JUnit 5
- Mockito (Mocking services)
- Testing APIs
- Integration testing
- Postman Automation (optional)

■ Module 13: Microservices Architecture (Optional)

- What are microservices?
- Monolithic vs Microservices
- Spring Cloud Basics
- API Gateway concept
- Service Registry (Eureka)
- Config Server basics
- Communicating between services (REST)

■ Module 14: DevOps (Optional)

Git & GitHub

- Branching
- Pull Requests
- Merge/Patch

Docker (Beginner Level)

- Dockerfile
- Dockerizing Spring Boot
- Docker Compose for app + DB

CI/CD

- GitHub Actions basic pipeline
- Auto deploy to cloud

■ Module 15: Cloud Deployment (Optional)

- Deploy Spring Boot on:
 - ✓ Render
 - ✓ Railway
 - ✓ AWS EC2 (intro)
- Deploy React/Angular on:
 - ✓ Netlify
 - ✓ Vercel
- Upload database to:
 - ✓ Render PostgreSQL
 - ✓ Railway MySQL

■ **Module 16: Performance Optimization**

- Hibernate optimization
 - Query optimization
 - Using DTOs
 - Caching basics
 - Using Postman for load testing
-

■ **Module 17: Capstone Full-Stack Project**

Students will build a real-world full-stack application:

Features:

- Authentication (JWT)
- Role-based access
- CRUD operations
- File uploads
- Dashboard UI
- Admin panel
- Search & filtering
- Pagination
- Deployed frontend + backend + DB

Project Examples:

- E-commerce Application
 - Hotel Booking System
 - Job Portal
 - Social Media Mini App
 - Real Estate Management System
 - Learning Management System (LMS)
-