Full Stack Web Development in Java – Overview

What is Full Stack Development?

Full Stack Development involves working on both the frontend (client-side) and backend (server-side) of a web application. In Java-based full stack development, Java is mainly used for server-side programming.

Tech Stack Breakdown

Frontend (Client-Side)

HTML5, CSS3, JavaScript – for UI structure, styling, and interactivity

Frameworks/Libraries:

React.js or Angular – for building dynamic, component-based UIs

Bootstrap or Tailwind CSS – for responsive design and layout

Backend (Server-Side)

Core Java – for logic implementation

Spring Framework (Spring Boot) – to create RESTful APIs and handle business logic

Hibernate or JPA – for object-relational mapping (ORM) and database interactions

RESTful Web Services – to enable communication between client and server

Database Layer

MySQL, PostgreSQL, or MongoDB – to store and manage application data

Development Tools and Platforms

IDEs: IntelliJ IDEA, Eclipse, or VS Code

Build Tools: Maven or Gradle

Version Control: Git and GitHub

Postman – for API testing

Deployment

Apache Tomcat – for local server deployment

Cloud Deployment: AWS, Heroku, or Azure

CI/CD Tools: Jenkins, GitHub Actions for automated build and deployment

Key Features of Java Full Stack Development

Strong type safety and performance from Java

Scalable architecture with Spring Boot

Clean separation of concerns using MVC pattern

Industry-standard practices like REST, ORM, and microservices