Java Full Stack Developer Curriculum

Module 1: Introduction to Web Development

- **Understanding Web Development**: Overview of web development, client-server architecture.
- **Front-end vs. Back-end**: Introduction to front-end technologies (HTML, CSS, JavaScript) and back-end technologies (Java, Spring, Databases).
- **Web Servers & Databases**: Introduction to web servers (Apache, Tomcat) and databases (MySQL, PostgreSQL, MongoDB).

Module 2: HTML Basics (Front-end)

- **Introduction to HTML**: Basic structure of an HTML document, the role of HTML in web development.
- **Formatting Tags**: Headings, paragraphs, text formatting (bold, italic), text alignment.
- **Lists**: Unordered, ordered, and definition lists.
- **Structural Tags**: div, span, header, footer, section, article, and nav.
- **Forms**: Input fields, radio buttons, checkboxes, dropdowns, and form validation.

Module 3: CSS Basics (Front-end)

- **Introduction to CSS**: Types of CSS - Inline, Internal, and External CSS.
- **Linking CSS to HTML**: How to link an external CSS file to an HTML page.
- **CSS Selectors**: Element, class, ID, and universal selectors.
- **Box Model**: Content, padding, border, margin, and box-sizing.
- **Text Styling**: Fonts, color, size, alignment, and decoration.

- **Background Styling**: Color, images, positioning, and repetition.

- **Layout Techniques**: Flexbox, Grid, float, and clear.

- **Positioning**: Static, relative, absolute, fixed, and z-index.

- **Display Properties**: Block, inline, inline-block, and none.

- **Responsive Design & Media Queries**: Introduction to responsive web design, breakpoints, and viewport adjustments.

Module 4: JavaScript Basics (Front-end)

- **Introduction to JavaScript**: Overview and role in web development.

- **Linking JavaScript to HTML**: Embedding JavaScript in HTML.

- **Variables & Data Types**: var, let, const, strings, numbers, Booleans, objects, arrays.

- **Operators**: Arithmetic, comparison, logical, and assignment operators.

- **Control Statements**: if-else, switch, ternary operators.

- **Arrays & Functions**: Array methods (map (), filter (), reduce ()), function declarations and arrow functions.

- **String Manipulation**: String methods (length (), slice (), split ()).

Module 5: Advanced JavaScript (Front-end)

- **Objects**: Key-value pairs and object manipulation.

- **ES6+ Features**: Let/const, template literals, DE structuring, spread/rest operators.

- **DOM Manipulation**: Accessing and modifying HTML elements.

- **OOP Concepts in JavaScript**: Classes, constructors, inheritance.

- **Asynchronous JavaScript**: Callback functions, Promises, and async/await.

- **Working with APIs**: Making HTTP requests using fetch ().

Module 6: Bootstrap (Front-end Framework)

- **Introduction to Bootstrap**: Overview of Bootstrap and its components.
- **Installation and Setup**: Including Bootstrap via CDN, downloading, or using npm.
- **Bootstrap Grid System**: Understanding rows and columns.
- **Common Components**: Navbar, Buttons, Cards, Alerts, Modals, Forms.
- **Customization**: Adjusting layouts, padding, margins, and overriding default styles with custom CSS.

Module 7: Introduction to Java (Back-end)

- **History and Evolution of Java**: Overview of Java's development, key features.
- **JVM, JRE, JDK**: Understanding the Java Virtual Machine, Java Runtime Environment, and Java Development Kit.
- **Writing a Simple Java Program**: Basic syntax and structure of a Java program.

Module 8: Core Java Concepts (Back-end)

- **Variables & Data Types**: Primitive and non-primitive types.
- **Control Statements**: if-else, loops, switch, break, continue.
- **Arrays & Collections**: Arrays, Lists, Sets, Maps.
- **Methods & Functions**: Method declaration, calling, parameters, overloading.
- **Object-Oriented Programming (OOP)**: Classes, objects, constructors, inheritance.

Module 9: Advanced Java (Back-end)

- **Polymorphism**: Method overriding, method overloading.
- **Encapsulation & Abstraction**: Using getters/setters, abstract classes, and interfaces.

- **Lambda Expressions & Streams API**: Functional programming techniques in Java.
- **Exception Handling**: try-catch-finally, custom exceptions.
- **Threads & Concurrency**: Creating threads, synchronization, thread lifecycle.

Module 10: Databases (Back-end)

- **Database Overview**: Introduction to relational databases, SQL.
- **SQL Basics**: DDL, DML, DQL, DCL, TCL commands.
- **CRUD Operations**: Insert, update, delete, select queries.
- **Joins and Aggregate Functions**: Inner join, outer join, group by, having, and aggregate functions.
- **Normalization**: 1NF, 2NF, 3NF, 4NF, and 5NF.
- **Database Constraints and Indexing**: Primary key, foreign key, unique constraints.

Module 11: Java Spring Framework (Back-end)

- **Introduction to Spring**: Overview of Spring Framework and its benefits.
- **Spring IoC (Inversion of Control)**: Dependency Injection and Bean Configuration.
- **Spring Boot**: Introduction to Spring Boot, setup, and project structure.
- **Spring MVC**: Model-View-Controller architecture, request handling.
- **Spring Data JPA**: Integrating Spring with JPA for database interaction.
- **Spring Security**: Basics of securing applications, authentication, and authorization.

Module 12: Hibernate ORM (Back-end)

- **Introduction to ORM**: Object-Relational Mapping, Hibernate framework.
- **Setting up Hibernate**: Configuring Hibernate with Java and Spring.
- **Hibernate Annotations**: Mapping Java objects to database tables.
- **CRUD Operations with Hibernate**: Insert, update, delete, and retrieve data.
- **Advanced Hibernate Concepts**: Caching, lazy and eager loading, transaction management.

Module 13: RESTful APIs Development (Back-end)

Introduction to REST: Representational State Transfer, RESTful principles.

- **Building REST APIs with Spring Boot**: Setting up a RESTful web service using Spring Boot.
- **RESTful CRUD Operations**: Create, Read, Update, Delete operations in REST.
- **Handling Request and Response**: Using @RequestMapping, @PathVariable, @RequestBody, and @ResponseBody.
- **Spring Security for APIs**: Securing REST APIs with JWT, OAuth2, and role-based access control.

Module 14: Front-end and Back-end Integration

- **Connecting Front-end and Back-end**: Using AJAX to connect JavaScript with Spring APIs.
- **Consuming REST APIs in JavaScript**: Fetching data from Spring Boot APIs using JavaScript.
- **Authentication**: Front-end login with back-end authentication using JWT.

Module 15: Full Stack Development with Java (Capstone Project)

- **Project Planning**: Defining the project requirements and scope.
- **Front-end Development**: Using HTML, CSS, JavaScript, and Bootstrap for the UI.
- **Back-end Development**: Setting up Java, Spring Boot, and Hibernate for the server-side logic and database.
- **Integrating Front-end and Back-end**: Connecting the front-end with back-end using REST APIs and handling security.
- **Testing and Debugging**: Unit testing, integration testing, and debugging.
- **Deployment**: Deploying the application on servers like Heroku, AWS, or locally using Docker.

Additional Skills

- **Version Control**: Using Git for version control and GitHub for collaboration.
- **Continuous Integration/Continuous Deployment (CI/CD)**: Automating build, testing, and deployment using Jenkins or GitLab CI.
- **Unit Testing**: Writing unit tests using JUnit and Mockito.
- **Agile Development**: Understanding Agile methodologies and Scrum framework.