

FULL STACK JAVA

CURRICULUM



1800-120-4748



www.teksacademy.com



support@teksacademy.com

Introduction to Full Stack Java Overview

- What is Full Stack Java?
- Why Full stack Java?
- Where is it used?
- Career Opportunities
- Learning Journey

Module 01: Core Java & Advanced Java

Introduction to Java

- Overview of Java Programming Language
- Features of Java (Object-Oriented, Platform-Independent, etc.)
- Java Development Kit (JDK) and Java Runtime Environment (JRE)
- Java Virtual Machine (JVM)
- Installing Java and Setting up Environment (PATH, CLASSPATH)

Java Basics

- Structure of a Java Program
- Compilation and Execution Process
- Data Types, Variables, and Constants
- Operators in Java
- Conditional Statements
- Control Flow Statements

Module 02: Object-Oriented Programming in Java

- iClasses and Objects
- Static Members
- Constructors
- This Keyword
- Inheritance (Single, Multilevel, Hierarchical, Multiple, Hybrid)
- super Keyword
- Polymorphism (Method Overloading and Method Overriding)
- Defining and Importing Packages
- Access Modifiers (private, public, protected, default)



- Encapsulation
- Interface Introduction
 - Implementing Multiple Inheritance using Interfaces
 - Difference between Abstract Classes and Interfaces
- Abstraction
- Nested and Inner Classes
- Final Classes

Module 03: Exception Handling

- Introduction to Exception Handling
- Try, Catch, Finally Block
- Throw and Throws
- Built-in Exceptions (Checked and Unchecked)
- Custom Exceptions

Multithreading in Java

- Creating Threads using Thread Class and Runnable Interface
- Thread Lifecycle and States
- Synchronization
- Inter-thread Communication (wait, notify, notifyAll)
- Deadlock
- Thread Pooling

Module 04. Java Collections Framework

- Introduction to Collection API
- Interfaces (List, Set, Map, Queue)
- Classes (ArrayList, LinkedList, HashSet, TreeSet, HashMap, TreeMap, LinkhashMap)
- Types of Linked Lists
 - Singly linked list
 - Doubly linked list
 - Circular linked list
- Iterators (Iterator, ListIterator)



- Comparable and Comparator Interfaces
- Collection Utility Methods

String Handling

- Immutable Strings and String Pool
- String Class and Methods
- StringBuffer and StringBuilder
- StringTokenizer

Input/Output (I/O) in Java

- Java I/O Streams (Byte Streams and Character Streams)
- Reading and Writing to Files
- File Input Stream
- File Output Stream
- Buffered Streams
- Serialization and Deserialization
- File Handling (File, FileWriter, FileReader)

Module 05: Advanced Java

Java Annotations

- Introduction to Annotations
- Built-in Annotations (@Override, @Deprecated, etc.)
- Custom Annotations
- Meta-Annotations
- Use of Annotations in Frameworks

Java 8 Features

- Lambda Expressions
- Functional Interfaces
- Stream API
- Method References
- Optional Class
- Default and Static Methods in Interfaces



Java Memory Management

- Stack vs Heap Memory
- Garbage Collection in Java
- Types of Garbage Collectors
- Finalize Method
- JVM Memory Architecture

Module 06: Java Design Patterns

- Singleton
- Factory design pattern
- DAO Pattern
- MVC Pattern

Advanced Topics

- Java Reflection API
- Dynamic Class Loading
- Java Concurrency Utilities (java.util.concurrent Package)
- Generics in Java
- Java Native Interface (JNI)

Module 07: SQL

Introduction to SQL

- History and evolution of SQL
- SQL vs NoSQL
- Types of databases (RDBMS, column-based, key-value, etc.)
- Database concepts: Tables, Rows, Columns, Relationships

SQL Data Types

- Numeric types (INT, FLOAT, DECIMAL)
- Character types (CHAR, VARCHAR, TEXT)
- Date and time types (DATE, TIME, TIMESTAMP)
- Boolean types
- BLOB (Binary Large Object)



Constraints in SQL

- PRIMARY KEY constraint
- FOREIGN KEY constraint
- UNIQUE constraint
- CHECK constraint
- DEFAULT constraint
- NOT NULL constraint

Database Design

- Normalization (1NF, 2NF, 3NF, BCNF)
- Denormalization
- Primary keys, foreign keys, and unique keys
- Indexing

SQL Commands

- DDL Commands
- DML Commands
- DQL Commands
- TCL Commands
- DCL Commands

Basic SQL Queries

- SELECT statement
- FROM and WHERE clause and logical operators (AND, OR, NOT)
- ORDER BY clause
- LIMIT and OFFSET clauses
- GROUP BY Clause
- HAVING Clause
- DISTINCT keyword

SQL Functions

- Aggregate functions (COUNT, SUM, AVG, MIN, MAX)
- Scalar functions (UPPER, LOWER, LENGTH, ROUND)
- Date functions (NOW, CURDATE, DATE_ADD, DATE_SUB)



- String Functions
- Control Functions
- Conversion Functions

Joins in SQL

- INNER JOIN
- LEFT JOIN (or LEFT OUTER JOIN)
- RIGHT JOIN (or RIGHT OUTER JOIN)
- FULL OUTER JOIN
- CROSS JOIN
- Self joins

Subqueries and Nested Queries

- Single-row subqueries
- Multi-row subqueries
- Correlated subqueries
- EXISTS and NOT EXISTS clauses

Set Operations

- UNION and UNION ALL
- INTERSECT
- EXCEPT (or MINUS)

Transactions in SQL

- ACID properties (Atomicity, Consistency, Isolation, Durability)
- COMMIT and ROLLBACK
- SAVEPOINT
- Transaction isolation levels (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE)

Indexes in SQL

- Purpose of indexes
- Types of indexes (single-column, multi-column)
- Unique and non-unique indexes



- Full-text index
- Index performance considerations

SQL Views

- Creating views
- Updating views
- Dropping views
- Advantages and limitations of views

Stored Procedures and Functions

- Creating stored procedures
- IN, OUT, and INOUT parameters
- Creating user-defined functions
- Differences between stored procedures and functions

Module 08: Java Database Connectivity (JDBC)

- Introduction to JDBC
- JDBC Architecture and Driver Types
- Connecting to Database
- Executing SQL Queries (Create, Read, Update, Delete)
- PreparedStatement, CallableStatement
- ResultSet and ResultSetMetaData
- Handling Transactions

Module 09: Servlets

Introduction to Servlets

- Definition and purpose of servlets
- Role of servlets in web applications
- Servlet vs CGI
- Java EE architecture overview



Servlet API Overview

- javax.servlet and javax.servlet.http packages
- Interfaces: Servlet, ServletRequest, ServletResponse, HttpServlet
- Servlet lifecycle methods: init(), service(), destroy()

Servlet Configuration and Deployment

- Servlet configuration in web.xml
- Annotations for servlet deployment (@WebServlet)
- Servlet context and servlet config
- Directory structure of a servlet-based web application

Handling Requests and Responses

- HTTP request-response model
- HTTP methods: GET, POST, PUT, DELETE
- Reading data from HttpServletRequest (headers, parameters)
- Writing response using HttpServletResponse (content type, status codes)

Session Management

- Importance of session management in web applications
- Techniques for session management:
 - Cookies
 - URL rewriting
 - Hidden form fields
 - HttpSession API (creating, tracking, invalidating sessions)
- Handling session timeout and lifecycle

Request Dispatching

- RequestDispatcher interface
- Forwarding requests to other resources (servlets, JSP, static files)
- Including response from other resources

Servlet Filters

- Definition and use of filters
- Filter lifecycle



- Applying filters to requests (doFilter())
- Chaining filters
- Filter configuration (using web.xml or annotations)

File Upload and Download in Servlets

- Handling file upload in servlets (MultipartConfig annotation, Part interface)
- Processing large file uploads
- Sending files for download (setting content type, handling large files)

Error Handling in Servlets

- Configuring error pages using web.xml
- Handling errors programmatically (error codes, exceptions)
- Custom error handling based on exceptions and HTTP status codes

Servlets with JDBC

- Establishing a database connection in servlets
- Performing CRUD operations with JDBC in servlets
- Best practices for handling database resources (connection pooling)

Interacting with JSP

- Introduction to JSP
- Uses of JSP
- Java Server package Vs Jakarta Server Page
- Role of JSP in MVC architecture
- Forwarding and including JSPs in servlets
- Sharing data between servlets and JSP (request, session, application scopes)

Module 10: Real Tools and Build Tools

- Servers
 - Tomcat Server
- IDE's
 - Eclipse



- Real tools
 - GIT/GITHUB
 - Maven/Gradle
 - Junit
- SDLC
 - What is SDLC
 - Types of phases in SDLC

Module 11: JAVA Spring Framework & Spring Boot Microservices

Introduction to Spring Framework

- Overview of Spring Framework
- Core features of Spring (Inversion of Control, Dependency Injection)
- Introduction to Spring Modules (Spring Core, Spring Data, Spring MVC, Spring Boot, etc.)

Spring Boot Overview

- What is Spring Boot?
- Spring Boot vs Spring Framework
- Advantages of using Spring Boot
- Spring Boot architecture and components
- Spring Boot CLI and IDE setup

Spring Boot Project Structure

- Understanding Spring Boot project structure
- Spring Boot starter templates
- Dependency management using Maven/Gradle
- Application properties and YAML configurations

Spring Boot Auto-Configuration

- Auto-configuration mechanism in Spring Boot
- Using annotations (@SpringBootApplication, @Component, @Bean)
- Customizing Spring Boot auto-configuration
- Auto wiring



Spring Data JPA and Persistence

- Overview of Spring Data JPA
- Entity relationships (One-to-One, One-to-Many, Many-to-Many)
- CRUD operations with Spring Data repositories
- Query methods and JPQL (Java Persistence Query Language)
- Pagination and sorting
- Transaction management with @Transactional

Spring Boot REST API Development

- Introduction to Postman Tool
- Building RESTful web services using Spring Boot
- Handling HTTP requests and responses (GET, POST, PUT, DELETE)
- RequestBody, ResponseEntity, and PathVariable
- Exception handling in Spring Boot (ControllerAdvice, ExceptionHandler)
- Validating REST API inputs using annotations

Spring Boot Security

- Introduction to Spring Security
- Configuring authentication and authorization
- Role-based access control (RBAC)
- JWT (JSON Web Token) integration with Spring Security
- OAuth2 for securing Spring Boot applications

Spring Boot Testing

- Unit testing with JUnit
- Testing Spring Boot applications using Spring Test
- Integration testing with Spring Boot
- Mocking dependencies with Mockito
- Writing test cases for REST controllers and services

Spring Boot with Databases

- Configuring Spring Boot with relational databases (MySQL, PostgreSQL, etc.)
- Database initialization and migrations using Flyway/Liquibase
- Connecting to NoSQL databases (MongoDB, Cassandra)
- Spring Boot with Redis for caching



Spring Boot and Messaging

- Introduction to messaging in Spring
- Integrating Spring Boot with messaging platforms (RabbitMQ, Kafka)
- Asynchronous messaging with Spring Boot
- Message converters and listeners

Introduction to Microservices Architecture

- What are microservices?
- Monolithic vs Microservices architecture
- Benefits and challenges of microservices
- Key concepts: API Gateway, Service Discovery, Circuit Breaker, Distributed Configuration

Spring Boot Microservices Development

- Building microservices using Spring Boot
- Communication between microservices (REST, messaging, gRPC)
- Service Registration and Discovery with Eureka
- Load balancing with Spring Cloud Ribbon
- API Gateway with Spring Cloud Gateway or Zuul
- Configuring externalized properties with Spring Cloud Config

Inter-Service Communication

- Synchronous communication with REST
- Asynchronous communication with Kafka, RabbitMQ
- Circuit Breaker with Spring Cloud Netflix Hystrix/Resilience4j
- Distributed tracing with Sleuth and Zipkin

Microservices Security

- Securing microservices with Spring Security
- OAuth2 and JWT-based authentication
- Securing API Gateway
- Implementing SSO (Single Sign-On) with Spring Boot



Spring Boot Microservices and Databases

- Handling databases in microservices (Database per service pattern)
- Transaction management in distributed systems
- Saga Pattern and Event-Driven Architecture
- Data consistency in microservices

Monitoring and Logging in Spring Microservices

- Centralized logging using ELK (Elasticsearch, Logstash, Kibana)
- Distributed tracing with Spring Cloud Sleuth and Zipkin
- Application metrics with Micrometer
- Integrating Prometheus and Grafana for monitoring

Deploying Spring Boot Microservices

- Introduction to Docker
- Introduction to Kubernetes
- Containerization with Docker
- Orchestrating containers with Kubernetes
- Using Helm for managing Kubernetes applications
- Cloud deployment with AWS/GCP/Azure
- CI/CD pipelines for Spring Boot microservices (Jenkins, GitLab CI, etc.)

IDE's

- Spring Tool Suit (STS)
- IntelliJ
- Eclipse

Embedded Servers

- Tomcat, Jetty, Undertow

Embedded Databases

- H2, HSQL, Derby

Tools

- POST MAN



Module 12: HTML

Introduction to HTML

- What is HTML?
- HTML and the World Wide Web
- Role of HTML in Web Development
- HTML Editors and Development Environment Setup
- Basic HTML Document Structure (DOCTYPE, <html>, <head>, <body>)

HTML Document Structure

- HTML Elements and Tags
- Block-level vs Inline Elements
- HTML Attributes (Global and Element-specific Attributes)
- Void Elements (e.g., ,
, <input>)

Text Formatting and Semantics

- Paragraphs, Headings, and Divisions
- Semantic HTML: <header>, <footer>, <article>, <section>
- Text-level elements: , , , etc.
- Lists: Ordered (), Unordered () Lists and Definition List (<dl>)
- Quotes: Blockquote and Inline Quotes

Links and Navigation

- Creating Hyperlinks with <a>
- Linking to External and Internal Resources
- Email Links, Telephone Links and Image Links
- Navigation Bars and Menus (with <nav>)

Images and Multimedia

- Inserting Images with
- Alt Attribute and Image Descriptions
- Responsive Images (<picture>, srcset)
- Embedding Audio (<audio>)
- Embedding Video (<video>)
- Using <iframe> for External Content (e.g., YouTube)



Tables

- Creating Tables: <table>, <tr>, <td>, <th>
- Table Headers, Footers, and Captions
- Colspan and Rowspan Attributes
- Table Accessibility Considerations

Forms and Input Handling

- Form Structure: <form>, action, method
- Common Input Types: Text, Password, Email, Number, Date, etc.
- Checkboxes, Radio Buttons, and Select Dropdowns
- Textarea and Submit Buttons
- Form Validation (Required Fields, Pattern Matching)
- Labeling Forms and Improving Accessibility

HTML5 Semantic Elements

- The Role of Semantic HTML in Modern Development
- New Structural Elements in HTML5 (<header>, <footer>, <main>, <aside>)
- Using <section> and <article> for Content Segmentation
- Benefits for SEO and Accessibility

Embedded Content

- Embedding External Resources with <iframe>
- Inline SVG Graphics
- Embedding External Stylesheets and JavaScript Files
- The <embed> and <object> Elements for External Applications (PDF, Flash)

Module 13: CSS

Introduction to CSS

- What is CSS?
- History and Evolution of CSS
- Advantages of CSS in web development



- Types of CSS: Inline, Internal, External
- Basic CSS Syntax and Structure
- CSS Selectors: Element, ID, Class, Universal, Grouping

CSS Box Model

- Understanding the Box Model
- Margins, Borders, Padding, and Content
- Box-sizing property

CSS Selectors in Depth

- Attribute Selectors
- Pseudo-Classes and Pseudo-Elements
- Combinators: Descendant, Child, Adjacent, General Sibling

CSS Layout Techniques

- Positioning: Static, Relative, Absolute, Fixed, Sticky
- Display Property: Block, Inline, Inline-Block, None
- Float and Clear
- CSS Flexbox: Introduction and Key Properties
- CSS Grid: Introduction and Key Properties

Typography in CSS

- Font Properties: Font-Family, Font-Size, Font-Weight, Font-Style
- Text Properties: Text-Align, Text-Transform, Text-Decoration, Line-Height
- Using Web Fonts

Styling Links and Lists

- Styling Hyperlinks: Link States
- Styling Ordered, Unordered, and Definition Lists

Colors, Backgrounds, and Borders

- Color Models: RGB, RGBA, HEX, HSL, HSLA
- Background Properties: Background-Color, Background-Image, Background-Position, Background-Repeat, Background-Attachment



- Border Properties: Border-Width, Border-Style, Border-Color, Border-Radius
- Gradients: Linear, Radial
- CSS Units and Values

CSS Units and Values

- Absolute Units: px, pt, cm, mm
- Relative Units: em, rem, vw, vh, %, fr
- Calculations using the calc() function

CSS Transitions and Animations

- CSS Transitions: Transition Properties, Timing Functions
- CSS Animations: Keyframes, Animation Properties

Responsive Design with CSS

- Media Queries: Breakpoints and Usage
- Viewport Meta Tag
- Responsive Units: %, vw, vh, rem, em
- Mobile-First Approach
- Flexbox and Grid for Responsive Layouts

CSS Variables (Custom Properties)

- Declaring and Using CSS Variables
- Scope and Inheritance of Variables

Browser Compatibility and Vendor Prefixes

- Handling Cross-browser Compatibility
- Vendor Prefixes for Different Browsers: -webkit-, -moz-, -ms-, -o-
- Tools for Compatibility Testing

Advanced CSS Features

- CSS Grid Advanced Techniques: Grid Areas, Template Layouts
- Advanced Flexbox Layout Patterns
- CSS Shapes and Masks
- CSS Clip-Path Property



- CSS Filters: Blur, Grayscale, Drop Shadows, etc.
- Advanced Selectors (Nth-child, Nth-of-type)

CSS for Web Accessibility

- Ensuring Text Readability and Color Contrast
- Focus and Active States for Keyboard Navigation
- CSS Guidelines for Accessible Web Design

CSS Grid vs. Flexbox

- When to Use Grid vs. Flexbox
- Differences and Use Cases

Module 14 : Javascripts

Introduction to JavaScript

- History and Overview
 - Brief history of JavaScript
 - ECMAScript and standardization
- Setting Up the Development Environment
 - Browsers and DevTools
 - Node.js setup (optional)
- Basic Syntax
 - Comments, variables, keywords
 - Data types and type conversion
 - Expressions and operators

JavaScript Fundamentals

- Variables and Scope
 - var, let, and const
 - Hoisting
 - Global, local, block scope
- Data Types
 - Primitive types: string, number, boolean, null, undefined, symbol, bigint
 - Complex types: object, array, function, array of objects



Type Conversion

- Implicit and explicit conversion
- typeof operator

Control Structures

- Conditionals
 - if, else if, else
 - Ternary operator
 - switch statement
- Loops
 - for, while, do...while
 - Iterating over objects and arrays (for...in, for...of)
 - break and continue

Functions

- Defining Functions
 - Function declarations and expressions
 - Arrow functions
 - Immediately Invoked Function Expressions (IIFE)
- Parameters and Arguments
 - Default parameters
 - Rest parameters and spread syntax
- Scope and Closures
 - Lexical scoping
 - Closures and practical use cases
- Callback Functions
 - Synchronous vs asynchronous callbacks

Object-Oriented Programming (OOP) in JavaScript

- Objects
 - Creating objects (object literals, new Object())
 - Accessing and modifying object properties



- Prototypes
 - Prototype chain
 - Prototypal inheritance
- Classes and Inheritance
 - Defining classes (class keyword)
 - Constructors
 - Class inheritance (extends, super)
 - Static methods and properties

Arrays and Advanced Array Methods

- Array Basics
 - Creating arrays, accessing elements
 - Array length, adding/removing elements
- Iterating Over Arrays
 - forEach(), map(), filter(), reduce(), some(), every(), sort()
- Array Mutability
 - Array methods that modify vs return new arrays
- Multi-dimensional Arrays
 - Working with nested arrays

Error Handling and Debugging

- Types of Errors
 - Syntax errors, runtime errors, logical errors
- Error Handling
 - try...catch block
 - finally statement
 - Throwing custom errors
- Debugging Tools
 - Using browser DevTools
 - Debugging with console methods (log, warn, error, time)

Asynchronous JavaScript

- Callbacks
 - Defining and using callbacks



- Promises
 - Creating and consuming promises
 - then(), catch(), and finally()
 - Promise chaining
- Async/Await
 - Writing asynchronous code with async and await
 - Error handling in async functions
- Event Loop
 - How JavaScript handles asynchronous operations
 - Microtasks and macrotasks

Document Object Model (DOM) Manipulation

- Understanding the DOM
 - DOM tree and nodes
- Selecting Elements
 - getElementById()
 - getElementsByTagName()
 - getElementsByClassName()
 - querySelector(), etc.
- Manipulating Elements
 - Changing content (innerHTML,.textContent)
 - Changing attributes, classes, styles
- Event Handling
 - Adding event listeners (click, keydown, mouseover, mouseleave)
 - Event delegation
 - Preventing default behavior

Browser APIs

- Timers
 - setTimeout(), setInterval()
- Local Storage and Session Storage
 - Storing and retrieving data
- Fetch API
 - Making HTTP requests



- Handling responses, JSON parsing
- Geolocation API
- Web Workers
 - Multithreading with web workers

Modular JavaScript

- Modules
 - ES6 modules (export, import)
 - Default and named exports
- CommonJS and AMD
 - require() and module.exports
- Bundlers
 - Using tools like Webpack or Parcel

Regular Expressions

- Basics of Regular Expressions
 - Syntax and pattern matching
- Common Methods
 - test(), exec()
 - String methods using regex (match(), replace())
- Flags and Modifiers

Module 15: jQuery

Introduction to jQuery

- What is jQuery?
- jQuery vs. JavaScript
- Advantages of using jQuery
- How to include jQuery in a project
 - CDN
 - Download and host locally

jQuery Selectors

- Overview of jQuery selectors
- Basic selectors (ID, class, element)



- Attribute selectors
- Hierarchical selectors
- Pseudo-selectors (e.g., :first, :last, :nth-child)

DOM Manipulation with jQuery

- Traversing the DOM (parent, child, sibling methods)
- Adding/removing elements
 - append(), prepend()
 - before(), after()
 - remove(), empty()
- Modifying element content
 - html()
 - text()
- Modifying element attributes
 - attr()
 - removeAttr()
 - prop()

jQuery Events

- Event binding methods
 - on(), off()
 - Shortcut methods (e.g., click(), hover(), submit())
- Event delegation
- Event propagation (bubbling and capturing)
- Preventing default behavior
- Stopping event propagation

jQuery Effects and Animations

- Showing and hiding elements
 - show(), hide()
 - toggle()
- Fading elements
 - fadeIn(), fadeOut()
 - fadeToggle()



- Sliding elements
 - `slideUp()`, `slideDown()`
 - `slideToggle()`
 - Custom animations using `animate()`
- Stopping animations
 - `stop()`
- Chaining animations

jQuery and Forms

- Selecting form elements
- Handling form events
 - `focus()`, `blur()`, `change()`
 - Form validation with jQuery
- Serializing form data
 - `serialize()`
 - `serializeArray()`

jQuery Utilities

- Working with `$.each()`
- Using `$.extend()`
- Utility functions (`$.trim()`, `$.isArray()`, etc.)
- Working with browser storage using `localStorage` and `sessionStorage`

Module 16: Bootstrap

Introduction to Bootstrap

- Overview of Bootstrap
- History and evolution of Bootstrap
- Importance of responsive design in web development
- Installation and setup of Bootstrap (via CDN, npm, or manual download)
- File structure of Bootstrap

Bootstrap Grid System

- Understanding the Bootstrap grid system
- Grid layout and breakpoints



- Building responsive layouts with the grid system
- Understanding container, row, and column classes
- Nesting grids and offsetting columns

Typography and Basic Elements

- Bootstrap's typography system
- Headings, paragraphs, and text utilities
- Lists, blockquotes, and code elements
- Inline elements and contextual text classes

Bootstrap Components

- Overview of Bootstrap components
- Buttons and button groups
- Forms: Form controls, input groups, layout options, and validation
- Navigation: Navbar, navs, and tabs
- Dropdowns and modals
- Alerts, badges, and breadcrumbs
- Cards and media objects

Utilities and Helpers

- Utility classes in Bootstrap
- Margin, padding, and spacing utilities
- Display and visibility classes
- Sizing utilities for width, height, and viewport settings
- Flexbox utilities for alignment, distribution, and order
- Text alignment and font utilities
- Background and color utilities

Advanced Components

- Carousel and image sliders
- Collapse and accordions
- Tooltips and popovers
- Pagination and progress bars
- Scrollspy and sticky navigation



Bootstrap Icons and Customization

- Introduction to Bootstrap Icons
- Adding and customizing Bootstrap Icons
- Customizing Bootstrap with Sass variables
- Overriding Bootstrap styles
- Creating custom themes with Bootstrap

Module 17: React JS

Introduction to React JS

- What is React?
- History and evolution of React
- Key features of React
- Understanding Single Page Applications (SPAs)
- React vs Other Frontend Frameworks (Vue, Angular)

Setting up the Development Environment

- Node.js and npm installation
- Installing React using Create React App (CRA)
- Project folder structure in React
- Overview of development tools (VS Code, React Developer Tools)

JSX (JavaScript XML)

- Introduction to JSX
- JSX vs HTML
- Embedding JavaScript expressions in JSX
- JSX attributes and children

Components in React

- Types of Components: Functional and Class-based
- Component lifecycle (Introduction)
- Creating and exporting components
- Component reusability



Props in React

- Passing data with props
- Default props
- Prop types (validating props)

State in React

- What is state in React?
- Managing local state in functional components
- The useState hook
- Updating and manipulating state

Event Handling

- Handling events in React
- Passing arguments to event handlers
- Synthetic events

Conditional Rendering

- Using if-else for conditional rendering
- Ternary operators and logical && for rendering

Lists and Keys

- Rendering lists in React
- Using keys in lists
- Handling dynamic data in lists

Forms in React

- Controlled vs Uncontrolled components
- Handling form inputs
- Form submission and validation

Lifting State Up

- Lifting state to a common ancestor
- Sharing state between components



React Router

- Introduction to React Router
- Setting up routing in a React application
- Route parameters and navigation
- Nested routes and redirection

React Hooks

- Introduction to Hooks in React
- useState, useEffect, useContext hooks
- Rules of Hooks
- Custom hooks and when to use them

Managing Side Effects with useEffect

- Introduction to side effects
- Fetching data with useEffect
- Cleaning up effects
- Dependency arrays in useEffect

Context API

- Introduction to React Context
- Creating a Context
- Providing and consuming context
- When to use Context vs props

Performance Optimization

- Introduction to React performance optimizations
- Memoization with React.memo and useMemo
- Reducing unnecessary re-renders
- Lazy loading with React.lazy and Suspense

Higher-Order Components (HOCs)

- Introduction to HOCs
- Creating and using HOCs
- Use cases for HOCs



Redux (State Management)

- Introduction to Redux
- Setting up Redux in a React application
- Actions, Reducers, and Store
- Connecting Redux to React components with react-redux
- Understanding the Redux flow

Deployment of React Applications

- Building a React application for production
- Hosting React apps on platforms like Netlify, Vercel, or GitHub Pages
- Optimizing bundle size and performance for deployment

Module 18: Material UI

Introduction to Material UI

- Overview of Material Design principles
- What is Material UI?
- Comparison of MUI with other UI libraries (Bootstrap, Ant Design, etc.)
- Installation & setup of Material UI in React applications

Theming with Material UI

- Default theme vs. custom themes
- Working with createTheme API
- Theme customization (colors, typography, spacing)
- Palette customization
- Dark mode integration

Components Overview

- Core Components:
 - Buttons, Icons, and Typography
 - Grid system (responsive design)
 - AppBar, Toolbar, and Drawer
- Form Components:
 - Text Fields, Select, Checkboxes, and Radios



Theming with Material UI

- Default theme vs. custom themes
- Working with createTheme API
- Theme customization (colors, typography, spacing)
- Palette customization
- Dark mode integration

Components Overview

- Core Components:
 - Buttons, Icons, and Typography
 - Grid system (responsive design)
 - AppBar, Toolbar, and Drawer
- Form Components:
 - Text Fields, Select, Checkboxes, and Radios
 - Autocomplete
 - Form validation & Form Control
- Feedback Components:
 - Alerts, Dialogs, Snackbars
- Navigation Components:
 - Tabs, Bottom Navigation, Breadcrumbs
 - Side Navigation (Drawer)

Layout and Styling

- Grid layout system and its breakpoints
- Box and Container components
- Flexbox utilities in Material UI
- Styling components with the sx prop
- Using makeStyles and styled API for custom styles

Icons and SVG Integration

- Using Material UI Icons
- Customizing Icon appearance
- Using third-party icons with MUI



Working with Forms

- Controlled vs. uncontrolled form components
- Validation using yup and formik
- Managing form states with Material UI

Responsive Design

- Breakpoints and media queries
- Responsive typography
- Responsive grids and containers
- Hidden components (for conditional rendering based on viewport size)

Data Display Components

- Tables (sorting, pagination, filtering)
- Cards (media and content)
- Lists and ListItem components
- Chips, Avatars, and Badges

Customization and Advanced Themes

- Overriding component styles
- Global style overrides
- Advanced theme options (shadows, transitions, z-index)

Customizing component variants

- Overriding component styles
- Global style overrides
- Advanced theme options (shadows, transitions, z-index)
- Customizing component variants

Routing and Navigation

- Integrating React Router with Material UI components
- Drawer, AppBar, and responsive layouts for navigation
- Breadcrumbs for hierarchical navigation



Animations and Transitions

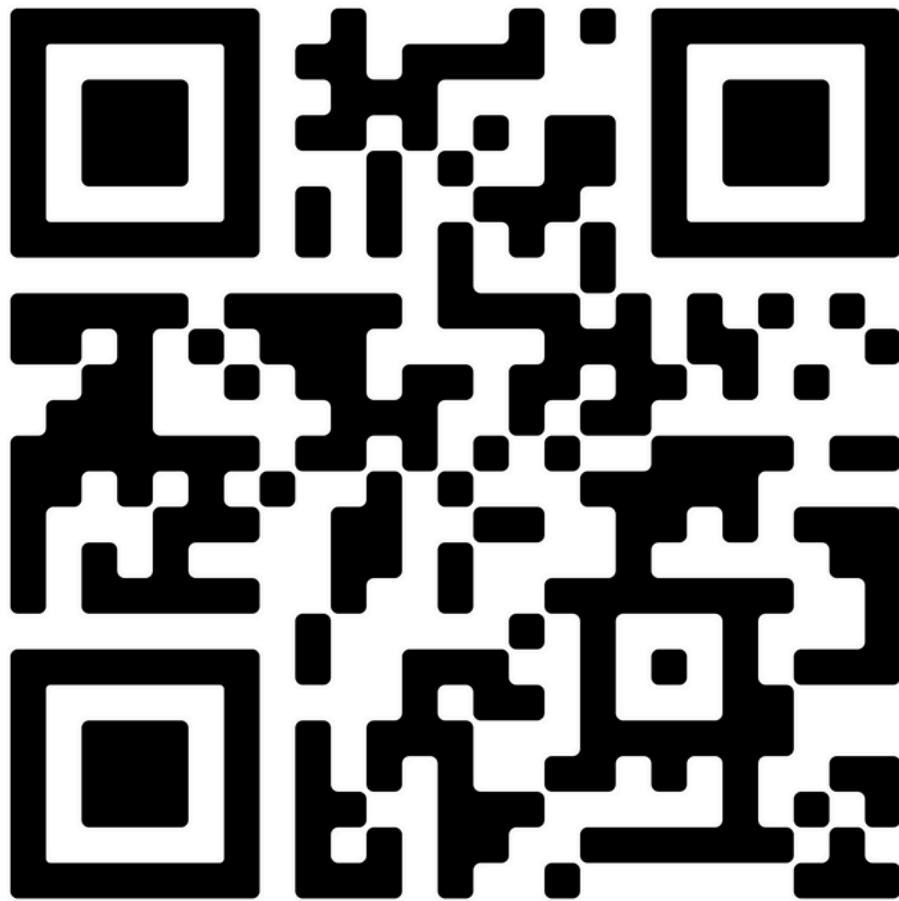
- Implementing transitions using MUI's Grow, Fade, and Slide
- Advanced animations using react-spring or framer-motion with MUI

Performance Considerations

- Optimizing Material UI for performance (tree-shaking, lazy loading)
- Using MUI's CssBaseline for consistent styling across browsers
- Reducing bundle size by importing components efficiently



Scan to view the Syllabus



Full Stack Java curriculum



1800-120-4748



www.teksacademy.com



support@teksacademy.com