



AI POWERED JAVA FULLSTACK COURSE



FRONTLINES EDUTECH PRIVATE LIMITED



ABOUT US

Frontlines Edutech Private Limited envisions bridging the gap between academia and industry. We understand the pain of unemployment and are trying to wipe off the words of impossible and unemployment from the brains of youth by training and turning them into diamonds. We provide tailor-made, hands-on, and need-based programs on par with industry standards. We believe that the youth of today is the future of tomorrow. And so our courses are tutored by experts from top industries who are passionate about mentoring the youth. Since its inception, Frontlines Edutech has earned the trust and gained recognition from thousands of learners across the state.



FOCUSED ON REFINING TALENT AND TURNING LEARNERS INTO HIGHLY SKILLED PROFESSIONALS SOUGHT AFTER BY TOP COMPANIES.



OUR MISSION AND VISION

At Frontlines Edutech Private Limited, our mission is to bridge the gap between academia and industry by offering practical, customized training programs that equip youth with the skills needed for easy employment. We aim to eliminate unemployment by fostering a mindset of possibility and transforming learners into skilled professionals. With expert industry mentorship, we empower the next generation with the tools to succeed in today's competitive world.

It is in our vision that each young person be prepared with the skill, confidence, and opportunity to succeed in their career. We want to lead an online skills development platform that provides focus on determining the needs of industries through education, so that it would be well-prepared in nurturing professionals who will shape the future of employment. Innovative training eliminates unemployment and helps in unlocking the potential of thousands of learners.





WHY CHOOSE US

We are Success Trainers, Mentors, and Motivators, passionate about helping you reach your full potential with a blend of warmth, wisdom, positivity, and proactivity. Our goal is always your highest good and greatest growth, offering support that is both caring and daring, fun and focused. We ensure learning is not only powerful but also enjoyable, making every step of your growth journey meaningful and engaging.





WHY LEARN THIS COURSE

- This course is for those who want to start their coding journey, whether a beginner or an expert.
- No prior programming knowledge is required.
- This course covers basics to advanced concepts straightforwardly and systematically.
- We take special care of Non-IT students
- Easy to learn
- Beginner Friendly
- The diversity of our learners adds richness to course discussions and interactions.





DELIVERABLES

8.PLACEMENT UPDATES

7.Q&A SESSIONS

6.COURSE COMPLETION CERTIFICATE

5.INTERVIEW GUIDANCE

4.LINKEDIN PROFILE BUILDING

2.RESUME BUILDING

3.DAILY ASSIGNMENTS

9.ON-DEMAND VIDEO COURSE CONTENT

10.DOWNLOADABLE RESOURCES

1.FROM SCRATCH TO MASTER LEVEL TRAINING



TOP COMPANIES HIRING

Infosys

tcs TATA
CONSULTANCY
SERVICES

accenture

cognizant

Capgemini

wipro

IBM

BOX8

KPMG



TOP CITIES FOR AI POWERED JAVA FULLSTACK DEVELOPER DEMAND

- BANGALORE (BENGALURU)
- GURGAON (GURUGRAM)
- HYDERABAD
- PUNE
- MUMBAI
- CHENNAI
- NOIDA

KEY SECTORS DRIVING DEMAND FOR AI POWERED JAVA FULLSTACK DEVELOPER

- TECHNOLOGY AND SOFTWARE DEVELOPMENT
- E-COMMERCE HEALTHCARE
- FINANCE AND FINTECH
- HEALTHCARE
- EDUCATION AND EDTECH
- DATA SCIENCE AND ANALYTICS
- ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
- MEDIA AND ENTERTAINMENT
- STARTUPS AND TECH INCUBATORS



AI POWERED JAVA FULLSTACK SYLLABUS

BACKEND

CORE JAVA

• Introduction to Java

- History and Features of Java
- JDK, JRE, JVM
- Setting Up Java Environment and Eclipse

• Basics of Java

- Structure of a Java Project
- Creating First Java Project
- Command Line Arguments
- Primitive Data Types
- Types of Variables
- Naming Conventions
- Operators
- Control Statements
- Loops
- Continue and Break Statements
- Type Casting
- String
- Arrays
- Methods in Java

• Object-Oriented Programming (OOP) Concepts

- Class and Object
- Constructor
- Constructor Chaining
- Static Keyword
- Inheritance
- This vs Super
- Polymorphism
- Abstraction
- Encapsulation
- Access Modifiers
- Interfaces

• Miscellaneous

- Garbage Collection
- File Handling
- Final Keyword
- Final vs Finalize

• Exception Handling

- Types of Exceptions
- Try, Catch, Finally
- Throw vs Throws
- Custom Exceptions

• Features of `java.lang` Package

- Object Class
- System Class
- Math Class
- Wrapper Classes
- String Buffer vs String Builder

• Multithreading

- What is a Thread in Java
- Thread Lifecycle
- Multithreading Advantages
- Creating Threads using Runnable Interface & Thread Class
- Synchronized Keyword

• Java Collections

- Collections Overview
- List, Set, Map Interfaces
- List Implementations
- Set Implementations
- Map Implementations
- Comparator and Comparable Interfaces

• Java 8 Features

- Lambda Expression
- Functional Interfaces
- Stream API
- Stream Filter
- Changes in Interfaces - Default and Static Methods
- forEach Method
- Optional Classes
- Date and Time API



DATABASE

- **Structured Query Language (SQL)**
 - MySQL Environment Setup
 - Introduction to RDBMS
 - DDL (Data Definition Language)
 - DML (Data Manipulation Language)
 - DQL (Data Query Language)
 - DCL (Data Control Language)
 - TCL (Transaction Control Language)
 - Commit and Rollback
 - Constraints (Unique Key, Not Null, Primary, Foreign Key)
 - Aggregate Functions (MIN, MAX, AVG, SUM, COUNT)
 - Joins
- **NoSQL**
 - MongoDB Environment Setup
 - Basic Overview of MongoDB
 - SQL vs NoSQL
 - Creating and Dropping Databases
 - Creating and Dropping Collections
 - Inserting, Updating, and Deleting Documents

JDBC

- **JDBC (Java Database Connectivity)**
 - Understanding Java Database Connectivity
 - Driver Manager
 - Connection
 - Statement
 - Prepared Statement
 - ResultSet
 - Result Set Metadata
 - CRUD Operations

HIBERNATE

- **Introduction to Hibernate**
 - What is ORM?
 - JDBC vs Hibernate
- **Hibernate Core Concepts**
 - Configuring Hibernate Configuration XML File
 - Understanding Entity Classes
 - Id Annotation and Generation Methods
 - Cascade Types
- **Manipulating and Querying Persistent Data**
 - Session Factory
 - Session
 - CRUD Operations
 - Transaction Management
- **Hibernate Relationships & Mapping**
 - One-to-One Mapping
 - One-to-Many Mapping
 - Many-to-One Mapping
 - Many-to-Many Mapping
 - Eager vs Lazy Loading
- **Advanced Hibernate**
 - Hibernate Query Language (HQL)
 - Native Queries
 - Named Queries

WEB APPLICATIONS

- Building a Dynamic Web Application
- Setting Up Tomcat Server
- Initializing a Servlet
- Servlet Life Cycle
- doPost and doGet Methods
- Overview of JSPs
- Directives and Expressions
- Using JSPs with Servlets



SPRING FRAMEWORK

- **Introduction to Spring**

- What is a Framework?
- Advantages of Spring
- Spring Modules Overview

- **Spring Core**

- What are Beans?
- Spring Containers
- Bean Lifecycle
- Understanding IOC & DI
- Bean Scope
- Singleton Design Pattern
- Spring Bean Configuration using XML
- Java-based Spring Bean Configuration
- Annotation-based Spring Bean Configuration
- Types of Dependency Injection

- **Spring JDBC**

- Configuring Data Source
- Configuring JDBC Template
- CRUD Operations using JDBC Template
- Row Mapper

- **Spring ORM**

- Advantages of Spring ORM over Hibernate
- Configuring LocalSessionFactoryBean
- Configuring HibernateTemplate
- CRUD Operations using HibernateTemplate

- **Spring MVC**

- Understanding MVC Architecture
- Dispatcher Servlet
- Types of Controllers
- Model and View
- Internal View Resolver
- Working with JSPs and HTML Pages

- **Spring AOP**

- Overview of Spring AOP
- Join Points, Point Cuts, Advice, Aspects
- Logging using AOP

SPRING BOOT

- **Introduction to Spring Boot**

- Features and Advantages of Spring Boot
- Spring vs Spring Boot
- Auto Configuration in Spring Boot
- Creating Spring Starter Projects
- Understanding application.properties and application.yml Files
- Spring Boot Starters
- Annotations in Spring Boot

- **Spring Boot Rest**

- Building Rest APIs
- RestController vs Controller
- Handling HTTP Methods
- Handling Status Codes using ResponseEntity
- Working with Thymeleaf

- **Spring Data JPA**

- Spring Data JPA Overview
- CRUD Operations with JPA Repository
- Finder Methods in JPA
- Java Persistence Query Language (JPQL)
- Custom Queries
- @Transactional Annotation

- **Exception Handling**

- RestControllerAdvice and ControllerAdvice
- Exception Handlers

- **Spring Boot Actuator**

- What is an Actuator?
- Metrics using Actuator
- Enabling Exposure Levels

- **Spring Security**

- Overview of Authentication
- Overview of Authorization
- Form Login using Spring Security
- In-Memory User Details Manager
- Managing User Credentials in Database
- Authentication using JWT



- **Spring AI**

- Overview of Spring AI
- What are Models?
- Use Cases of AI in Spring Applications
- Setting up a Spring Boot Project with AI Capabilities
- Using OpenAI API in Spring Boot
- Introduction to LLMs
- Implementing Chatbots with Spring AI
- Using Prompt Templates in Spring AI
- Generating Images using DALL-E

UNIT TESTING

- **JUnit**

- Importance of Unit Testing in Software Development
- Setting Up JUnit in Java
- Annotations in JUnit
- Assertions in JUnit
- What is Mockito?
- Mocking Database using Mockito
- Performing Tests on Service Layers

GIT & GITHUB

- Setting Up Git Environment
- What is a Version Control System?
- Why Use Git?
- Different Stages in Git
- Cloning a Repo using Git Clone
- Commands in Git
- Viewing Commit History
- What is Branching?
- Creating, Deleting & Switching Branches
- Creating a Pull Request
- Reviewing and Merging Pull Requests
- Resolving Merge Conflicts

MICROSERVICES

- **Introduction to Microservices**

- Monolithic vs Microservices Architecture
- Advantages & Challenges of Microservices

- **Service Discovery**

- What is Service Discovery?
- Creating Eureka Server
- Registering Services in Eureka

- **Inter-Service Communication**

- Synchronous vs Asynchronous Communication
- Configuring Rest Template
- Communicating using Rest Template
- Configuring Feign Client
- Communicating using Feign Client

- **Config Server**

- What is a Config Server?
- Creating an In-Memory Config Server
- Config Server with GitHub

- **API Gateway**

- What is a Gateway?
- Setting Up an API Gateway
- Routing Requests to Services

- **Circuit Breaker**

- Why Do We Need a Circuit Breaker?
- Setting Up Resilience4j
- Writing Fallback Methods
- Setting Thresholds and Managing Circuit



FRONTEND

HTML

• How the Internet Works

- History of Web
- How computers communicate
- How computers send data globally
- Domain Name, IP & MAC Addresses
- How ISP and DNS work

• Client-Server Architecture

- Client-Server Model
- Difference between Client and Server
- HTTP Request and Response Cycle
- What happens when you visit a website
- Front-end vs Back-end
- Static vs Dynamic Websites
- Web Hosting
- Installing and set up

• HTML Basics

- Understanding HTML
- Creating First HTML Page
- HTML Structure and Tags
- Tag vs element

• HTML Tags

- Working with text elements Heading tag
- Paragraph tag
- Image tag
- Br,hr,sub,sup,bold,italic,span tags
- Lists,ordered and unordered lists
- Using HTML attributes
- Navigation using anchor tags
- Commenting in HTML

• List and Media Tags

- List
- Ordered list

- Unordered list
- Media tags
- Audio
- Video Tags

• HTML Forms and Inputs

- Importance of Forms
- Creating a Simple Form
- Types of Input Fields
- File input type
- Email input type
- Password input type
- Submit button input type many more
- Browser Tools
- Forms creations
- Signup and login forms creation
- Semantic Tags

CSS

• Basics of CSS

- Introduction to CSS
- Syntax and Selectors
- Adding CSS to HTML
- Difference Between Selectors
- Styling Text
- Comments

• Styling with CSS

- Working with Colors
- CSS Units
- Box Model
- Padding
- Margin
- Border Properties
- Background Properties
- Shadow Property
- Text Properties
- Font Properties

• Advanced CSS Concepts

- Display Properties
- Flexbox for Alignment



- Grid System
- Positional Properties
- Overflow
- **CSS Enhancements**
 - Pseudo Classes and Elements
 - CSS Transitions and Animations
- **Responsive Web Design**
 - Mobile-First vs Desktop-First
 - Measurement Units
 - Viewport Meta Element
 - Media Queries
 - Responsive Images and Typography
- **Bootstrap**
 - What is Bootstrap?
 - Using bootstrap
 - Container layout, buttons, badges
 - Alert, Button Group
 - Navbar
 - Card components
 - Form layout

JAVASCRIPT

- **JavaScript Basics**
 - Introduction to JavaScript
 - Linking JavaScript Files
 - Running JavaScript in Browser
 - Variables and Keywords
 - Logging with JavaScript
 - Working with Strings
 - JavaScript Data Types
 - Operators
 - Variable Hoisting
- **Loops & Conditionals in JavaScript**
 - Conditional Operators
 - Loops
 - Recursion
 - Loop Control Statements

- **Functions in JavaScript**
 - Understanding Functions
 - Function Parameters
- **Arrays and Objects in JavaScript**
 - Creating and Accessing Arrays
 - Array Methods
 - Iterating Over Arrays
 - Working with Objects
 - Object Methods
 - Timing Events
- **Document Object Model (DOM) Manipulation**
 - Introduction to DOM
 - Understanding DOM Structure
 - Fetching Elements
 - Manipulating DOM Elements
 - Creating and Removing DOM Elements
- **Event Handling in JavaScript**
 - Event Handling
 - Scroll Events, Mouse Events, Key Events
 - Form Handling
 - Browser Events
- **Browser Functionalities in JavaScript**
 - Browser Object Model
 - Web APIs
- **Client-Side Storage**
 - Introduction to Client-Side Storage
 - Cookies
 - Local Storage
 - Session Storage
 - Comparison of Cookies, Local Storage, and Session Storage
 - Security considerations in frontend storage



- **Object-Oriented Programming in JavaScript**
 - Classes and Objects
 - Constructor Functions and Prototypes
 - Class Inheritance
- **Asynchronous JavaScript**
 - Understanding Asynchronous Code
 - Event Loop
 - Callbacks
 - Promises
 - Async/Await
 - JavaScript Timers
- **Error Handling in JavaScript**
 - Types of Errors
 - Using try-catch
 - Creating Custom Errors
 - Error Handling in Asynchronous Code

REACT

- **Introduction to React**
 - What is React
 - History of React and Origin
 - React Components
 - Single Page Applications
 - Virtual DOM
 - NPM and Installing Packages
 - Folder Structure
 - Setting Up React
 - Writing markup in jsx
- **Styling in React**
 - Component-Based Styling
 - CSS Modules and Styled Components
 - Dynamic Styling
 - Responsive Design
 - Animations and Transitions

- **React Core Components**
 - Creating Functional Components
 - Using CSS in React
 - React State
 - Props and Conditional Rendering
 - Rendering and Filtering Arrays
 - useState Hook
 - setState
 - Stateful Components
 - Stateless Components
- **React Routing**
 - Basic Routing
 - Passing Params



PROJECTS

MAJOR PROJECTS

- 1 . MedSync - AI-Powered Hospital Management System
- 2 . StayEase - Accommodation Management System
- 3 . Smart Bank Management System
- 4 . PFMS-Personal Finance Management System
- 5 . Spotify Clone – Music Streaming UI
- 6 . TicketEase - Ticket Management System
- 7 . NewsBlog Application

MINI PROJECTS

- 1 . Music Player Application
- 2 . Online Food Ordering System
- 3 . Gated Community - Complaint Tracker Application
- 4 . AI-Based Resume Screening System using Spring AI
- 5 . WeatherNow - Weather Application
- 6 . Personal Portfolio Website
- 7 . Product Showcase Carousel



ALL THE BEST





FRONTLINES EDUTECH

Building Trust & Careers

Follow Us



FRONTLINES MEDIA



FRONTLINES EDUTECH



FRONTLINES MEDIA-IGNITED-MINDS



FRONTLINES MEDIA



FRONTLINES EDUTECH



For Enquiry Contact Us:

📞 +(91)-83330 77727

✉️ media.frontlines@gmail.com

🌐 www.frontlinesedutech.com