Java Full Stack Development with AI Integration - Course Syllabus

Greens Technologies - Building end-to-end applications with Java, Spring Boot and practical AI. 32 modules, 10 AI topics, 1 mini project.

Course overview

This course takes you from programming basics to AI-enabled full stack applications: frontend, Core Java, Oracle SQL, Spring Boot, enterprise Java, tools, and OpenAI / Gemini integration. It has 32 modules plus 10 AI topics, ending in a mini project.

Why learn Java Full Stack?

End-to-end development across frontend, backend and databases, Enterprise-ready backend with Java and Spring Boot, Real workflow: APIs, security, testing and version control, Project-based learning that connects every module into complete applications.

Why add AI to Java applications?

Add chat, summarisation, document processing and intelligent search, Use OpenAI / Gemini APIs from Java and Spring Boot, Learn secure API handling, validation and error management, Build practical features that fit existing business applications.

Course roadmap

Stage 1 - Modules 1-2: Frontend Foundations: HTML, CSS, Bootstrap, JavaScript
Stage 2 - Module 3: Frontend Framework: React.js, components, hooks, routing, API calls
Stage 3 - Modules 4-12: Core Java: OOP, collections, exceptions, files
Stage 4 - Modules 13-20: Database and JDBC: Oracle SQL, joins, views, JDBC
Stage 5 - Modules 21-26: Spring Ecosystem: Spring Core, DI, Spring Boot, REST APIs, JPA
Stage 6 - Modules 27-30: Enterprise Development: Security, Hibernate, Microservices, JUnit
Stage 7 - Modules 31-32: Tools and Workflow: Git, GitHub, Maven
AI Stage - AI Topics 1-10: AI Integration: OpenAI / Gemini, Spring AI, chatbot, mini project

Part 1 - Frontend Development (3 Modules)

Course objective

Build responsive interfaces for Java-backed web applications.

1. HTML, CSS and Bootstrap

HTML5: basic structure, semantic tags, forms, tables, lists, links and multimedia, CSS3: selectors, box model, Flexbox, positioning and responsive media queries, CSS animations: transform, translate and keyframes, Bootstrap components, grid system, Flexbox utilities, layouts and breakpoints

2. JavaScript

Variables, data types, operators and control flow, Functions, arrays, array methods and object manipulation, DOM manipulation and event handling, Asynchronous programming: callbacks, promises and async/await, JavaScript modules, forms, routing and higher-order functions

3. React JS

JSX, functional and class-based components, props and lists/keys, Lifecycle methods, conditional rendering and event handling, Hooks: useState, useEffect, useRef, useContext and useReducer, React Router, API integration and data fetching using Axios

Frontend hands-on practice

Create a responsive page using semantic HTML and CSS, Build a form with validation and event handling, Create reusable React components and pass data using props, Manage component state and navigate between pages, Fetch data from a REST API and display loading, success and error states.

Greens Technologies

Suggested frontend projects

BMI calculator, To-Do application (create, update, track tasks), registration form validation, e-commerce interface with product listing and cart, weather dashboard using API data.

Outcome

Students can create responsive interfaces and connect them to backend APIs.

Part 2 - Core Java (9 Modules)

Course objective

Foundations, object-oriented programming and advanced programming concepts for backend development.

4. Introduction to Java

Overview, history and features of Java, JVM, JRE and JDK; platform independence and Write Once, Run Anywhere, Java development environment and IDE setup, Structure of a Java program, naming conventions, input and output, Packages, class accessibility and package access
LAB: Write, compile and run a first Java program

5. Data Types and Basic Syntax

Primitive and non-primitive data types, Data type declaration, ranges and calculations, Wrapper classes, Scanner class: importing, creating Scanner objects and input methods
LAB: Small programs that accept input and process different data types

6. Object-Oriented Programming

Classes, objects and methods, Four pillars of OOP: inheritance, polymorphism, abstraction and encapsulation, Inheritance: types, superclass, subclass, extends and implements, Polymorphism: method overloading, method overriding and upcasting
LAB: Model a simple real-world entity using classes and objects

7. Abstraction, Encapsulation and Access

Abstraction: abstract classes, abstract methods and interfaces, Encapsulation and data hiding, Getters, setters, POJO / Bean classes and implementation, Access specifiers and access modifiers: types, scope and use with inheritance

8. Control Statements and Strings

Decision making: if, if-else, nested if-else, if ladder and switch, Loops: for, nested for, while and do-while, Jump statements: break and continue, String methods, immutability, string types and memory, StringBuilder and StringBuffer

9. Arrays and Collections

Single- and multi-dimensional arrays: declaration, initialization and length, Array properties and limitations, Collections Framework overview, hierarchy and generics, List, Set and Map interfaces: hierarchy, declaration, methods and implementation comparison, Iterating through a Map and common collection operations

10. Constructors and Variables

Introduction to constructors and types of constructors, Constructor chaining and constructors in inheritance, this and super keywords in constructors, Variable scope, lifetime and access modifiers, Final variables and static final variables

11. File Operations

File class and file operations, Creating directories and subdirectories, Reading, writing, copying and moving files, File operations with directories, file exceptions and permissions

Greens Technologies

12. Exception Handling

Introduction to exceptions and types of exceptions, Handling exceptions using try-catch, Finally block, Throwing and declaring exceptions
LAB: Write programs that validate input and handle expected failures

Part 3 - Database: Oracle SQL and JDBC (8 Modules)

Course objective

Query, manage and retrieve relational data.

13. SQL and Database Fundamentals

What SQL is, its history, importance, syntax and structure, Database, database vs file system and types of databases, DBMS features and components; RDBMS and DBMS comparison

14. SQL Commands and Data Types

DDL: CREATE, ALTER, DROP and TRUNCATE, DQL: SELECT; DML: INSERT, UPDATE and DELETE, DCL: GRANT and REVOKE; TCL: COMMIT, ROLLBACK and SAVEPOINT, Numeric, string, date/time, binary and other data types

15. Constraints and Filtering

NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK and DEFAULT, Indexes, WHERE clause and comparison operators, BETWEEN, IN, LIKE; AND, OR and NOT

16. Aggregation and Window Functions

COUNT(), SUM(), AVG(), MIN() and MAX(), GROUP BY, HAVING vs WHERE and grouping by multiple columns, ROW_NUMBER(), RANK(), DENSE_RANK(), LEAD() and LAG(), PARTITION BY vs GROUP BY

17. Single Row Functions

String functions: UPPER, LOWER, LENGTH, SUBSTRING and REPLACE, Numeric functions: ABS, ROUND, CEIL, FLOOR and POWER, Date functions and date extraction, Conversion functions such as CAST and CONVERT where supported

18. Joins and Subqueries

INNER, LEFT, RIGHT, FULL, SELF and CROSS JOIN, Single-row, multi-row and correlated subqueries, Subqueries in SELECT, FROM and WHERE, Subqueries compared with joins

19. Set Operators, ORDER BY and Views

UNION, UNION ALL, INTERSECT and MINUS (EXCEPT in some databases), Differences between joins and set operators, ORDER BY with ascending/descending order, multiple columns and expressions, Creating, updating and dropping views; types, advantages and use cases

20. JDBC

Introduction to JDBC, its purpose and components

Part 4 - Spring Framework and Spring Boot (6 Modules)

Course objective

Build maintainable Java backend applications and REST services.

21. Spring Core and Beans

Spring Framework overview, features, benefits and architecture, Spring Bean and bean lifecycle, Bean configuration using XML, annotations and Java configuration, Dependency Injection: purpose, benefits, constructor, setter and field injection

Greens Technologies

22. Configuration and Database Basics

Annotation-based configuration and advantages over XML, @Configuration, @Bean and @ComponentScan, Spring Data, DataSource and database connection configuration, application.properties / application.yml, JDBC URL and credentials, JdbcTemplate: overview, benefits, setup and core methods, Support concepts: @Qualifier with @Autowired; logging with Log4j (dependencies, configuration, loggers and appenders)

23. Introduction to Spring Boot

Overview, benefits and key features of Spring Boot, Spring Boot compared with the traditional Spring Framework, Creating a project using Spring Initializr or manually, Project directory structure, starters, dependencies and configuration files, Running a Spring Boot application and creating a simple RESTful web service

24. CRUD Application Development

Creating a Spring Boot project with CRUD operations, Model/entity class, repository layer, service layer and controller layer, Database configuration and connection setup, Implementing CRUD operations with database integration

25. Spring Boot with JPA

Introduction to JPA and its role in database operations, Benefits of using JPA with Spring Boot, Defining entity classes and implementing CRUD operations

26. Exception Handling in Spring Boot

Importance of exception handling in web applications, @ExceptionHandler and handling specific controller exceptions, Custom exceptions and mapping exceptions to HTTP status codes

Part 5 - Enterprise Java: Security, ORM, Microservices and Testing (4 Modules)

27. Spring Security

Overview of Spring Security, Authentication, authorization, principal and roles, Configuring Spring Security in a Spring Boot application, Role-Based Access Control (RBAC), Spring Security filters and Security Filter Chain, Securing REST APIs and stateless authentication with JWT

28. Hibernate and JPA

Hibernate overview, purpose and comparison with JDBC, Object-Relational Mapping (ORM) basics and advanced Hibernate features, Hibernate Entity Manager and differences between Hibernate and JPA, Persistence Context: transactional and extended types, SessionFactory, Hibernate Sessions and their relationship, Configuration using hibernate.cfg.xml and programmatic Configuration, Creating and closing SessionFactory; obtaining Session instances

29. Microservices

Definition and characteristics of microservices, Monolithic architecture compared with microservices, Benefits, challenges and appropriate use cases, Designing microservices and communication between services, Secure API gateways and real-world implementation concepts, Migration strategies from monolithic applications

30. JUnit

Benefits of unit testing and features of JUnit, JUnit compared with other testing frameworks, JUnit setup, Maven dependencies and first test, JUnit annotations and parameterized tests, Dependency injection in test methods and mocking, Integration with build tools and advanced JUnit features

Greens Technologies

Part 6 - Tools and Workflow (2 Modules)

31. Git, GitHub and Bitbucket

Git basics, installation and terminology: repository, commit and branch, Creating, switching and merging branches, Workflow models: centralized, feature branch, Gitflow and forking, Pull requests, remotes, push, pull and fetch, GitHub basics and differences between Git and GitHub

32. Maven

Maven features, benefits and installation, Maven project structure and conventions

Part 7 - AI Integration: Industry Practical Concepts (10 AI Topics)

Course objective

Add real AI capabilities to Java and Spring Boot applications using OpenAI and Gemini APIs, through beginner-friendly tasks that reflect common company workflows.

AI 1. Introduction to AI Integration in Java

What AI integration means: send a request to an AI service and use the response, Business use cases: customer support, document workflows, knowledge search and business automation, Application flow: Java / Spring Boot backend -> AI provider -> application response, Prompts, tokens, models and responses in simple terms, Provider selection: capabilities, pricing, usage limits and privacy considerations
LAB: Make a first AI API request from a Java application and display the response

AI 2. OpenAI API and Gemini API using Java

Understand API keys, endpoints, request payloads and response formats, Send requests from Java using an HTTP client, Parse JSON responses and handle errors, timeouts and rate limits, Compare provider responses using the same sample prompt, Keep credentials in environment variables or secure configuration
LAB: Call an AI API from a Java service and compare sample responses

AI 3. Spring Boot Integration with AI APIs

Create a Spring service class for AI requests, Expose AI features through Spring Boot REST endpoints, Configure provider details, timeouts and request validation, Connect a frontend or API client to the AI-enabled backend
LAB: Build a Spring Boot endpoint that returns an AI-generated response

AI 4. Building AI-Powered Chatbot Applications

Chat request/response flow and conversation context, System prompts and simple assistant roles such as customer support or HR helpdesk, Connect a chat interface to a Spring Boot backend, Handle long answers, loading states, failures and conversation history
LAB: Create a basic customer-support chatbot backed by a Spring Boot API

AI 5. AI-Based Text Generation and Summarization

Prompt basics: clear instructions, context, examples and output format, Generate draft emails, descriptions, notes and business text, Summarise long text and documents into key points, Rewrite, translate and correct text; validate generated content
LAB: Build a text summarisation endpoint for user-submitted content

AI 6. Document Ingestion and Indexing

Load and extract text from supported PDF, DOCX, HTML and other permitted sources, Split extracted content into smaller, meaningful text chunks, Generate embeddings for chunks and retain source text and metadata, Store vector embeddings in a vector database for later retrieval, Understand offline ingestion, indexing and update workflows
LAB: Build an ingestion pipeline that extracts, chunks, embeds and indexes sample documents

Greens Technologies

AI 7. Semantic Search and Retrieval

Convert a user query into an embedding using an embedding model, Search indexed vectors using similarity measures such as cosine similarity, Retrieve relevant chunks and metadata from the vector database, Augment the prompt with retrieved context before sending it to an LLM, Generate a context-grounded response and return it to the application
LAB: Create a semantic-search endpoint that retrieves relevant document content for a user query

AI 8. Introduction to Spring AI

Overview of Spring AI and its role in Java applications, Understand model providers, prompts and model interactions, Use framework abstractions to connect an AI provider, Compare direct API integration with framework-based integration
LAB: Create a small Spring AI proof-of-concept using a configured provider

AI 9. REST API, Database and Security

Design REST endpoints that invoke AI services and validate requests, Connect AI features with JPA/Hibernate repositories where required, Store permitted prompts, results and application records, Protect API keys; use authentication, authorization and usage limits, Handle timeouts, quota limits, invalid responses and safe error messages, Avoid sending sensitive user data without authorization; review AI outputs
LAB: Secure an AI endpoint and test success, invalid input and provider failure

Document ingestion and semantic search - end-to-end workflow

Offline - document ingestion and indexing

Steps: 1. Source documents (PDF, DOCX, HTML, databases, APIs), 2. Document loader / extractor, 3. Text splitter / chunker, 4. Embedding model (e.g. text-embedding-ada-002, sentence-transformers), 5. Vector embeddings (original text + source info) stored in a vector database (e.g. Pinecone, Weaviate, Chroma).

Online - query processing and response generation

Steps: 1. User query, 2. Embedding model converts the query to a vector, 3. Vector search / retrieval (nearest neighbour, cosine similarity) returns the top-k relevant chunks and metadata, 4. Prompt augmentation: "Answer the question based on the context below: [Context] Question: [Query]", 5. Large language model (e.g. GPT-4, Claude, Llama 2) generates the response, 6. Generated response is returned to the user.

AI 10. Mini Project: AI-Powered Customer Support Portal

Build a full-stack support portal where users submit questions and receive AI-generated suggested answers.

Required features

1. Frontend: support request and chat interface, 2. Backend: Spring Boot REST API with controller, service and repository layers, 3. AI: integrate OpenAI or Gemini through a backend service, 4. Database: save support requests and reviewed responses where appropriate, 5. Security: protect API credentials and restrict access to support records, 6. Testing: validate normal responses, invalid input, timeout and provider errors.

Build stages

1. Plan the features, screens, data model and application architecture, 2. Create the frontend and Spring Boot project, 3. Connect the database using JPA/Hibernate, 4. Integrate the AI provider through a service layer, 5. Add validation, authentication and exception handling, 6. Test the complete workflow and publish source code to GitHub.

Greens Technologies

Learning outcomes

Build responsive frontend applications using HTML, CSS, JavaScript and React, Develop Java backend applications with Spring Boot and REST APIs, Connect relational databases using SQL and JPA/Hibernate, Apply security, testing and Git-based development practices, Integrate AI-powered chat, text summarisation, document ingestion and semantic search, Demonstrate a full-stack application with practical AI functionality.

Program Details - Tools & Technologies

Frontend

HTML5, CSS3, Bootstrap, JavaScript, React.js, Axios

Backend

Core Java, Spring Core, Spring Boot, REST APIs, Spring Security (JWT), Microservices

Database & ORM

Oracle SQL, JDBC, JPA, Hibernate

Testing & tools

JUnit, Git, GitHub, Bitbucket, Maven, Log4j

AI

OpenAI API, Gemini API, Spring AI, embeddings, vector database (Pinecone / Weaviate / Chroma)

Learn | Build | Get Hired - Greens Technologies.

Greens Technologies