AI LABS & RESEARCH CENTER

Full Stack Developer Program — Detailed Course Syllabus

MindForge AI Labs & Research Center offers an industry-aligned Full Stack Developer Program. All learners begin with a Common Foundation covering programming basics, OOPs, DSA, and frontend technologies — then specialize into either Track 1 (.NET) or Track 2 (Java) for backend mastery, concluding with a real-world capstone project.

Program Duration: 14–16 Weeks Total (Tentative) — Common Foundation: 6–8 Weeks + Track 1 / Track 2: 6–8 Weeks | **Mode:** Classroom / Online | **Format:** Live projects + Practical assessments

Common Foundation — Shared by Both Tracks

Every learner completes this foundation before choosing a specialization. It covers core programming, OOPs, data structures & algorithms, and the shared frontend stack (HTML, CSS, Bootstrap, JavaScript, React JS) used in both tracks.

Total Duration: 6–8 Weeks (Tentative)

Foundation 1 — Programming Basics: C & C++

Duration: 1–1.5 Weeks

- C fundamentals: variables, data types, operators, control statements
- Functions, recursion, pointers, arrays, strings
- Structures, unions, file handling in C
- C++ basics: classes, objects, constructors/destructors
- Function & operator overloading, references vs pointers
- Memory management: new/delete, stack vs heap

Foundation 2 — Object-Oriented Programming (OOPs) Concepts

Duration: 0.5–1 Week

- Four pillars: encapsulation, abstraction, inheritance, polymorphism
- Classes, objects, access modifiers
- Method overloading vs overriding
- Abstract classes vs interfaces
- Real-world OOP modeling exercises (language-agnostic, applied later in C#/Java)

Foundation 3 — Data Structures & Algorithms (DSA)

Duration: 1.5–2 Weeks

- Arrays, strings, linked lists (singly, doubly, circular)
- Stacks, queues, hashing/hash maps
- Trees: binary tree, BST, traversals, AVL basics
- Graphs: BFS, DFS, shortest path basics
- Sorting algorithms: bubble, selection, insertion, merge, quick sort
- Searching algorithms: linear, binary search

- Time & space complexity (Big-O notation)
- Recursion & backtracking, basic dynamic programming
- Problem-solving practice for technical interviews

Foundation 4 — HTML5 & CSS3

Duration: 0.5–1 Week

- Semantic HTML5 elements, forms, tables, media tags
- CSS3 fundamentals: box model, selectors, positioning
- Flexbox & CSS Grid layouts
- Responsive design principles, media queries
- CSS transitions, transforms, and animations

Foundation 5 — Bootstrap

Duration: 0.5–1 Week

- Bootstrap grid system & responsive breakpoints
- Bootstrap components: navbar, cards, modals, forms, buttons
- Utility classes for spacing, alignment, and typography
- Customizing Bootstrap themes
- Building responsive layouts rapidly with Bootstrap

Foundation 6 — JavaScript Essentials

Duration: 1–1.5 Weeks

- Variables, data types, operators, control flow
- Functions, closures, callbacks, ES6+ features (let/const, arrow functions, destructuring, spread/rest)
- DOM manipulation & event handling
- Asynchronous JavaScript: promises, async/await, fetch API

Foundation 7 — React JS

Duration: 1–1.5 Weeks

- React fundamentals: JSX, components (class & functional), props, state
- React Hooks: useState, useEffect, useRef, useContext, useMemo, useCallback, useReducer
- React Router DOM for client-side routing
- Form handling and validation
- API integration using Axios — CRUD operations
- State management with Redux Toolkit
- Component lifecycle and performance optimization
- Styling with CSS / Bootstrap in React projects

Track 1 — .NET Backend Specialization

Builds on the Common Foundation with C#, ASP.NET Core, EF Core, and Clean Architecture for enterprise-grade backend systems.

Total Duration: 6–8 Weeks (Tentative)

Module 1 — C# & .NET Fundamentals

Duration: 1–1.5 Weeks

- Introduction to .NET Framework, .NET Core/.NET, CLR, MSIL, JIT compilation
- CLS, CTS, CLI and platform independence
- C#-specific OOP implementation: classes, objects, constructors, destructors
- Static vs dynamic binding, sealed/partial keywords
- Exception handling: try/catch/finally, custom exceptions, exception hierarchy
- Collections: List, Dictionary, Stack, Queue, LinkedList, SortedList
- Generics and type safety
- Multithreading & async/await, Task Parallel Library, thread synchronization

Module 2 — Advanced .NET

Duration: 1–1.5 Weeks

- Properties, indexers, delegates & events
- Anonymous types, lambda expressions, extension methods
- Reflection basics
- LINQ: filtering, sorting, aggregation, joins, deferred execution, IEnumerable vs IQueryable
- Serialization & deserialization: JSON, XML, Binary formatters
- I/O streams: file read/write, byte/line operations
- Design patterns commonly used in enterprise .NET apps (Repository, Singleton, Factory)

Module 3 — Database & ADO.NET / EF Core

Duration: 0.5–1 Week

- SQL Server fundamentals: DDL, DML, joins, stored procedures, triggers, functions
- ADO.NET connected layer: Connection, Command, DataReader
- ADO.NET disconnected layer: DataAdapter, DataSet, DataTable
- Entity Framework Core: Code-First & Database-First approaches
- EF Core migrations, query filters, multi-tenant data isolation
- Connection pooling and performance considerations

Module 4 — ASP.NET Core & MVC

Duration: 1–1.5 Weeks

- ASP.NET Core architecture, middleware pipeline, dependency injection
- MVC pattern: routing, controllers, actions, action results
- Razor views, view models, partial views, layouts
- Model binding & data annotations, validation
- Session, cookies, state management
- Authentication & authorization basics (Identity, JWT)

Module 5 — Web API & Clean Architecture

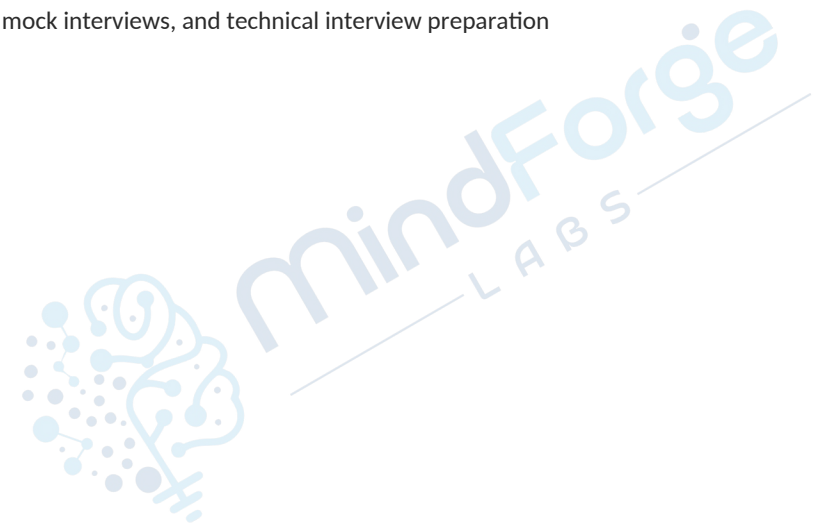
Duration: 1–1.5 Weeks

- RESTful API design principles, HTTP verbs, status codes
- Building Web APIs with ASP.NET Core, Swagger/OpenAPI documentation
- Clean Architecture layering: Domain, Application, Infrastructure, API
- CQRS pattern with MediatR, request/response pipelines
- Repository & Unit of Work patterns
- Multi-tenant SaaS design: tenant resolution, EF Core global query filters
- Background jobs with Quartz.NET / hosted services
- Third-party integrations (e.g., Twilio-style external APIs)

Module 6 — Real-World Project, Git & Deployment

Duration: 1–1.5 Weeks

- Git & GitHub: branching strategy, pull requests, code reviews
- Capstone project: end-to-end full stack application (ASP.NET Core API + React frontend)
- Unit testing basics with xUnit/NUnit
- CI/CD pipeline setup using GitHub Actions
- Deployment to IIS on a Windows VPS / cloud hosting basics
- API testing with Postman, debugging common HTTP errors
- Resume building, mock interviews, and technical interview preparation



Track 2 — Java Backend Specialization

Builds on the Common Foundation with Core/Advanced Java, Spring Boot, Spring Security, and Hibernate for enterprise-grade backend systems.

Total Duration: 6–8 Weeks (Tentative)

Module 1 — Core Java

Duration: 1–1.5 Weeks

- Java basics: JDK/JRE/JVM, variables, data types, naming conventions
- Java-specific OOP implementation: classes, objects, packages
- Static vs final vs this/super keywords, access modifiers
- String handling: String, StringBuffer, StringBuilder, immutability
- Exception handling: checked/unchecked exceptions, custom exceptions
- I/O: FileReader/Writer, Streams, Scanner class, serialization/deserialization
- Collections Framework: List, Set, Map implementations, Queue, Generics
- Multithreading: thread lifecycle, synchronization, thread priority, daemon threads

Module 2 — Advanced Java (JDBC, Servlets, JSP)

Duration: 1–1.5 Weeks

- JDBC API: DriverManager, Connection, Statement, PreparedStatement, ResultSet
- Servlets: lifecycle, Generic vs HTTP Servlet, request/response handling
- Session tracking: cookies, hidden form fields, URL rewriting, HttpSession
- Filters: lifecycle, authentication filters
- JSP: lifecycle, scripting elements, implicit objects, directives, action elements
- Expression Language (EL) and JSTL
- MVC architecture in JSP/Servlet applications
- Building a database-driven registration/CRUD example

Module 3 — Spring Core, Spring MVC & Spring Boot

Duration: 1–1.5 Weeks

- Introduction to Spring Framework and its modules
- Inversion of Control (IoC) and Dependency Injection (constructor/setter)
- Bean lifecycle, bean scopes, autowiring
- Spring MVC: request flow, controllers, views, form data binding
- Introduction to Spring Boot: auto-configuration, starter dependencies
- Building REST APIs with Spring Boot

Module 4 — Spring Security & Hibernate (ORM)

Duration: 1–1.5 Weeks

- Spring Security basics: authentication & authorization, roles
- Hibernate architecture and configuration (XML & annotation-based)
- Hibernate mappings: one-to-one, one-to-many, many-to-many
- Inheritance mapping strategies: Table Per Hierarchy/Concrete/Subclass
- HQL and Criteria API, transaction management
- Hibernate caching basics

Module 5 — Database & Build Tools

Duration: 0.5–1 Week

- SQL fundamentals: queries, joins, procedures
- Database connectivity using JDBC drivers
- Maven: project structure, pom.xml, dependency management, build lifecycle

Module 6 — Real-World Project, Git & Deployment

Duration: 1–1.5 Weeks

- Git & GitHub: branching, pull requests, collaborative workflow
- Capstone project: end-to-end full stack application (Spring Boot API + React frontend)
- API testing with Postman
- Application packaging and deployment basics (Tomcat/cloud)
- Resume building, mock interviews, and technical interview preparation

Program Summary

- **Common Foundation: 6–8 Weeks (Tentative)**
- **Track 1 (.NET) or Track 2 (Java): 6–8 Weeks (Tentative)**
- **Total Program Duration: 14–16 Weeks (Tentative)**

Program Outcomes — Both Tracks

- Strong foundation in programming, OOPs, and data structures & algorithms
- Build and deploy a production-grade full stack web application from scratch
- Hands-on Git/GitHub workflow, code reviews, and CI/CD basics
- REST API design, authentication, and database integration
- Resume building, mock interviews, and placement support

Who Should Enroll

Freshers, computer science graduates, and working professionals looking to transition into full stack development. No prior coding experience required for the Common Foundation; a willingness to practice problem-solving is helpful.