

60-DAY INSTRUCTOR-LED PROGRAM · 4 MODULES · 8 EXERCISES

Full-Stack .NET 8 + React 18

A structured, production-grade curriculum taking you from **C# fundamentals** through a **deployed full-stack application**. One focused hour per day. SOLID principles from Day 1.

60

DAYS

4

MODULES

8

EXERCISES

1 hr

PER DAY

.NET 8 Web API

EF Core 8

SQL Server

React 18

TypeScript

Redux Toolkit

JWT Auth

Tailwind CSS

xUnit / Jest

Playwright E2E

IIS + Vercel

GitHub Actions

BACKEND

C# 12 · .NET 8 Web API · EF Core 8 · SQL Server · Serilog · Redis Caching · JWT + Refresh Tokens

FRONTEND

React 18 · TypeScript · Redux Toolkit · RTK Query · React Hook Form · Tailwind CSS · Vite

DEVOPS & QA

Git · GitHub Actions CI/CD · xUnit · Jest · RTL · Playwright · IIS · Vercel

WHAT YOU WILL BUILD

Four Comprehensive **Modules**

MODULE 1 · DAYS 01–15

C# Fundamentals & Design Patterns

Master the language and patterns that underpin every enterprise .NET application — OOP, LINQ, async/await, and the four canonical design patterns applied to real-world scenarios.

- OOP: Classes, Interfaces, Inheritance, Polymorphism
- Collections, Generics & LINQ (filtering, joins, grouping)
- async / await, Task, CancellationToken, ValueTask
- Repository, Unit of Work, Strategy & Observer patterns
- SOLID Principles & Clean Code review
- 3 Graded Exercises

MODULE 2 · DAYS 16–35

.NET Core 8 Web API

Build a fully secured, production-quality REST API — EF Core migrations, JWT auth with refresh tokens, AutoMapper DTOs, global error handling (RFC 7807), caching, rate limiting, and Swagger.

- Controllers, Routing, DI & Service Lifetimes
- EF Core 8 — CRUD, Relationships, Migrations
- JWT Authentication & Role-Based Authorization
- Middleware, Serilog, API Versioning, Swashbuckle
- Caching (IMemoryCache/Redis), Pagination, Security
- 4 Graded Exercises + Performance & Deploy Prep

MODULE 3 · DAYS 36–52

React 18 Frontend & Design Patterns

Build a professional SPA with Hooks, React Router v6, Redux Toolkit + RTK Query, React Hook Form, and advanced patterns (HOC, Compound Components, Render Props). Add Jest & RTL tests.

- JSX, Props, State, Events & Core Hooks
- React Router v6, Protected Routes & JWT flow
- Redux Toolkit — Slices, Thunks & RTK Query
- Advanced Patterns: HOC, Render Props, Compound
- Performance (useMemo, useCallback, React.memo)
- 1 Graded Exercise + Jest & RTL testing

MODULE 4 · DAYS 53–60

Full-Stack Integration & Capstone Project

Assemble every skill into a deployable product — CORS, end-to-end testing with Playwright, IIS / Self-Hosted backend, Vercel frontend, and a final live code-review presentation.

- CORS configuration & full-stack project bootstrap
- Architecture design, ERD & wireframing
- Backend + Frontend scaffolding & JWT E2E
- End-to-End Testing with Playwright / Cypress
- Deployment: IIS (API) + Vercel/Netlify (React)
- Final Presentation, Code Review & Retrospective

|| *Every session enforces **SOLID principles, Clean Code practices, and RESTful naming conventions.** Students write production-quality code from **Day 1** — not toy examples, not scaffolded starters.*

DAY TOPIC & KEY CONCEPTS

MODULE 1 — C# FUNDAMENTALS & DESIGN PATTERNS
WEEK 1 · DAYS 01–05

- 01 **C# Syntax & Data Types**
 Variables, value vs. reference types, string interpolation, nullable types, constants, type casting
- 02 **Control Flow & Methods**
 if / switch expressions, loops, method overloading, optional params, ref/out, expression-bodied members
- 03 **OOP — Classes, Objects & Constructors**
 Class anatomy, auto-properties, object initializers, static vs. instance, records
- 04 **OOP — Inheritance & Polymorphism**
 virtual/override/sealed, abstract classes, constructor chaining, is/as operators
- 05 **OOP — Interfaces & Abstraction** ★ Ex 1
 Interface contracts, explicit impl, default interface methods, dependency inversion

WEEK 2 · DAYS 06–10

- 06 **Collections & Generics** ✓ Rev 1
 List<T>, Dictionary, HashSet, Queue, Stack, IEnumerable<T>, generic constraints
- 07 **LINQ — Filtering, Ordering & Projection**
 Where, Select, OrderBy, Skip/Take, deferred vs. immediate execution
- 08 **LINQ — Aggregation, Joins & GroupBy**
 GroupBy, Join, GroupJoin, SelectMany, Zip, composite keys
- 09 **Exception Handling & Debugging**
 try/catch/finally, custom exceptions, when filters, AggregateException, global handlers
- 10 **Asynchronous Programming — async/await** ★ Ex 2
 Task, async/await, CancellationToken, WhenAll/WhenAny, ValueTask, deadlock avoidance

WEEK 3 · DAYS 11–15

- 11 **Design Patterns — Singleton & Factory** ✓ Rev 2
 Thread-safe Singleton, Factory Method, Abstract Factory; testability considerations
- 12 **Design Patterns — Repository Pattern**
 Generic IRepository<T>, EF-backed implementation, separation of data access
- 13 **Design Patterns — Unit of Work**
 IUnitOfWork, coordinating repositories, transaction management, EF Core integration
- 14 **Design Patterns — Strategy & Observer**
 IStrategy interchangeable algorithms, IObserver<T>/IObservable<T>, event-driven design
- 15 **SOLID Principles & Clean Code Review** ★ Ex 3
 SRP, OCP, LSP, ISP, DIP with real refactoring examples; meaningful names, small functions

MODULE 2 — .NET CORE 8 WEB API
WEEK 4 · DAYS 16–20

- 16 **.NET Core Intro & Project Structure** ✓ Rev 3
 SDK CLI, Program.cs minimal hosting model, launchSettings, appsettings, middleware overview
- 17 **Controllers, Routing & Action Results**
 ApiController, attribute routing, route constraints, IActionResult, HTTP status helpers
- 18 **Dependency Injection & Service Lifetimes**
 Singleton/Scoped/Transient, options pattern, IConfiguration, keyed services (.NET 8)
- 19 **EF Core — Setup & Migrations**
 DbContext, code-first migrations, Fluent API, seed data, migration commands
- 20 **EF Core — CRUD Operations** ★ Ex 4
 Add, Find, Update, Remove, SaveChanges, AsNoTracking, async CRUD, ChangeTracker

WEEK 5 · DAYS 21–25

- 21 **EF Core — Relationships & Navigation** ✓ Rev 4
 One-to-many, many-to-many, Fluent API, eager/lazy loading, Include/ThenInclude
- 22 **RESTful API Design & HTTP Methods**
 REST constraints, resource naming, GET/POST/PUT/PATCH/DELETE semantics, idempotency
- 23 **Model Validation & Data Annotations**
 Required, Range, StringLength, custom attributes, IValidatableObject, FluentValidation
- 24 **Middleware Pipeline & Custom Middleware**
 Use/Run/Map, IMiddleware, exception middleware, correlation ID, request timing
- 25 **JWT Authentication & Authorization** ★ Ex 5
 JWT structure, AddJwtBearer, token generation, refresh tokens, claims-based identity

WEEK 6 · DAYS 26–30

26

Role-Based Authorization & Claims ✓ Rev 5

Roles vs. claims vs. policies, IAuthorizationHandler, resource-based authorization

27

AutoMapper & Data Transfer Objects

DTO rationale, AutoMapper profiles, ProjectTo<T>, ReverseMap, CQRS-aligned models

28

Global Error Handling & Structured Logging

ProblemDetails (RFC 7807), IExceptionHandler (.NET 8), Serilog, log levels, sinks

29

API Versioning & Swagger / OpenAPI

Asp.Versioning.Mvc, URL/header versioning, Swashbuckle, JWT security definition

30

Repository + Unit of Work with EF Core ★ Ex 6

Full generic repo inside a real API, UoW wrapping DbContext, service orchestration

WEEK 7 · DAYS 31–35

31

Caching Strategies ✓ Rev 6

IMemoryCache, IDistributedCache, Redis, cache-aside pattern, invalidation

32

Background Services & Hosted Services

IHostedService, BackgroundService, Channel<T>, worker service, graceful shutdown

33

Filtering, Sorting & Pagination

PagedResult<T>, IQueryable composition, cursor-based pagination, dynamic sorting

34

API Security Best Practices

HTTPS/HSTS, rate limiting (ASP.NET 8), CORS policies, secrets management

35

Performance Optimization & Deployment Prep ★ Ex 7

Response compression, output caching, health checks, publish profiles, CI/CD intro

DAY TOPIC & KEY CONCEPTS

MODULE 3 — REACT 18 FRONTEND & DESIGN PATTERNS
WEEKS 8–9 · DAYS 36–45

- 36 **React Setup, JSX & Component Basics** ✓ Rev 7
Vite scaffold, JSX rules, functional components, props, component composition
- 37 **Props, State & Event Handling**
Prop types, controlled inputs, synthetic events, lifting state, one-way data flow
- 38 **useState & useEffect Hooks**
State initialisation, effect cleanup, dependency arrays, stale closures
- 39 **useContext & Custom Hooks**
Context API, createContext/useContext, custom hook extraction, hook composition
- 40 **Axios Integration & API Calls** ★ Ex 8
Axios instance, interceptors, async data fetching, loading & error states
- 41 **React Router v6 & Navigation** ✓ Rev 8
BrowserRouter, Routes/Route, useNavigate, useParams, nested routes, lazy loading
- 42 **Forms & React Hook Form**
Controlled vs. uncontrolled, RHF validation rules, error messages, field arrays
- 43 **State Management with Context API**
Global state design, context providers, useReducer, avoiding prop drilling
- 44 **Redux Toolkit — Slices, Thunks & RTK Query**
createSlice, createAsyncThunk, RTK Query endpoints, cache management
- 45 **Higher-Order Components & Render Props**
HOC factory, Render prop signature, cross-cutting concerns, comparison with hooks

WEEKS 10–11 · DAYS 46–52

- 46 **Compound Components Pattern**
Implicit state sharing via context, flexible APIs (Select/Option, Tabs/Tab)
- 47 **Error Boundaries & React Suspense**
Class-based error boundaries, fallback UI, Suspense for lazy imports & data
- 48 **Performance Optimization**
useMemo, useCallback, React.memo, virtualization, profiler, avoid re-renders
- 49 **Authentication Flow in React**
JWT storage, PrivateRoute HOC, token refresh interceptor, session timeout
- 50 **Responsive Design — Tailwind CSS**
CSS Modules scope, Tailwind utilities, responsive breakpoints, dark mode
- 51 **Testing — Jest & React Testing Library**
render, screen, fireEvent, findBy queries, mocking Axios, accessibility assertions
- 52 **Review & Integration Prep**
Full-stack recap, CORS, proxy, API integration patterns, dev-to-prod checklist

MODULE 4 — FULL-STACK INTEGRATION & CAPSTONE
CAPSTONE SPRINT · DAYS 53–60

- 53 **CORS Configuration & Full-Stack Bootstrap**
Named CORS policies, preflight requests, Vite proxy setup, solution folder structure
- 54 **Capstone Planning — Architecture & ERD**
Domain modelling, ER diagram, API contract (OpenAPI spec), React component tree
- 55 **Backend API Scaffolding & Core Endpoints**
EF Core migrations, generic repo, UoW, core CRUD endpoints, seed data
- 56 **React Frontend Scaffolding**
Feature-folder structure, shared UI, RTK Query API slice, routing setup
- 57 **JWT Authentication End-to-End**
Login flow: React form → .NET API → JWT issued → Axios interceptor → protected routes
- 58 **End-to-End Testing with Playwright**
Playwright test runner, page object model, login / CRUD E2E tests, CI integration
- 59 **Deployment — IIS + Vercel / Netlify**
Publish .NET API to IIS (self-contained), deploy React to Vercel, environment vars, HTTPS
- 60 **Final Presentation, Code Review & Retrospective**
Live demo, architecture walk-through, instructor feedback, peer review, certificates

TECH STACK

.NET 8

EF Core 8

SQL Server

React 18

TypeScript

Web API

ORM

Database

Frontend SPA

Typed JS

Tailwind CSS

Styling

Redux Toolkit

State Mgmt

JWT Auth

Security

xUnit / Jest

Testing

Playwright

E2E Testing

Serilog

Logging

Swashbuckle

OpenAPI

Git / GitHub

VCS

GitHub Actions

CI/CD

IIS / Vercel

Deployment

PREREQUISITES

HARDWARE & OS

- ✓ Windows 10/11 — 8 GB RAM min (16 GB recommended)
- ✓ 30 GB free disk space for tools, SDKs & projects

SOFTWARE

- ✓ Visual Studio 2022 Community (ASP.NET & Node.js workloads)
- ✓ VS Code + Prettier, ESLint, Tailwind IntelliSense
- ✓ Node.js LTS (v20+), .NET 8 SDK, Git for Windows
- ✓ SQL Server 2022 Developer + SSMS 20

KNOWLEDGE

- ✓ HTML5 & CSS3 basics (structure, selectors, Flexbox)
- ✓ Basic programming logic — variables, loops, conditions
- ✓ Comfort using a terminal / command-line

ASSESSMENT & GRADING

Weekly Exercises (8 total)	40 %
Module Quizzes (4 total)	20 %
Capstone Project	40 %

CAPSTONE SCORING

- 25 pts** Clean, SOLID-compliant backend architecture
- 20 pts** Secure JWT auth flow (end-to-end)
- 20 pts** React component design & state management
- 20 pts** Full-stack data flow — EF Core → API → React
- 15 pts** Deployment & CI/CD pipeline

Passing Criteria

Score ≥ 70/100 overall · Each exercise ≥ 50% to proceed

Ready to enroll?

Start with a **free first session** — no commitment. One hour per day is all it takes.
rightstepsolutions.in · info@rightstepsolutions.in