TR-01-L2: Modern Full Stack Engineering (Advanced)

From Builder to Architect - Lead Production Systems with Confidence

You can build apps. Now learn to build them at scale. This advanced track transforms solid full-stack developers into engineers who can architect, optimize, and lead production systems. In 14 weeks, you'll master monorepos, NestJS, microservices, caching, Docker, observability, and advanced AI integration - the skills that separate senior engineers from junior ones.

The focus is on production engineering judgment, system design thinking, and the ability to lead technical decisions in real teams.

Why This Course?

The Market Reality

Global Context: Senior full-stack engineers are valued for their ability to design systems, lead implementation, and make sound architecture decisions under real constraints. Companies increasingly need engineers who can balance delivery speed, reliability, and maintainability across growing products.

Nepal Context: Kathmandu's growing startup ecosystem - including FinTech (F1Soft, Khalti), EdTech (Saarathi, Gurubaa), and HealthTech - needs engineers who can architect scalable systems, lead teams, and make infrastructure decisions.

Your Opportunity: This course positions you for **senior developer, tech lead, and architect roles** - both locally and internationally. You'll graduate with production-grade projects demonstrating advanced engineering capability.

Nepal-Relevant Reality	Opportunity
Senior engineers are scarce in Nepal	You fill the most valuable gap in the market
Startups need architects, not just coders	Lead system design and technical decisions
Remote senior roles pay significantly more	Access international compensation from Nepal
Production systems require observability and scale	Build the skills most engineers never develop

Course Snapshot

Parameter	Details
Course Code	TR-01-L2
Title	Modern Full Stack Engineering (Advanced)
Duration	3.5 Months (14 Weeks)
Schedule	Monday to Friday (Mon–Fri, 5 Days/Week), 2 Hours/Day
Total Hours	140 Hours of Live Training
Batch Size	Maximum 10 Students
Course Fee	NPR 36,000
Prerequisites	TR-01-L1 completion or equivalent. Prior React, backend, and SQL experience. Comfort building and deploying web applications. Saarathi Gate Assessment (diagnostic, no pass/fail) before Day 1.
Self-Study	Minimum 2 hours/day outside class (mandatory)
Outcome	Senior Full Stack Engineer / Tech Lead

Your Learning Week

Day	Activity
Mon–Fri	2-hour live class session (hands-on, project-based)
Mon–Fri	Minimum 2 hours self-study & practice (mandatory)
Saturday	No classes - flexible self-study, peer collaboration, project work
Sunday	Whole day self-learn time. Classrooms remain fully open for you to come in, study, collaborate with peers, and build projects. (Highly recommended for networking!)

> *Every student MUST spend at least 2 dedicated hours a day on focused coding/practice beyond the classroom at home. This is non-negotiable for success, it is what separates graduates who get hired from those who don't.*

Week-by-Week Curriculum

Phase 1: Advanced Frontend, Data Workflows & Testing (Weeks 1–3, 3 Weeks, 30 Hours)

Week	Focus Area	What You'll Master
Week 1	L1 Foundations Recap & Advanced Engineering Thinking	L1 foundations recap (JS, React, Node), self-assessment, architecture thinking shift, advanced tooling
Week 2	Data & State Patterns	Forms, async state, real-time flows, query management
Week 3	Testing & Code Quality	E2E coverage, mocking, CI checks, maintainable test design

Phase 2: Monorepos, NestJS, Messaging & Database Scaling (Weeks 4–7, 4 Weeks, 40 Hours)

Week	Focus Area	What You'll Master
Week 4	Turborepo & Monorepos	Shared packages, app boundaries, repo structure, build flows
Week 5	NestJS	Modules, DI, guards, pipes, service structure
Week 6	Microservices & Queues	Service boundaries, events, RabbitMQ, background-job thinking
Week 7	Database Scaling	Query tuning, search, replicas, high-load trade-offs

Phase 3: Caching, Performance, Docker & Observability (Weeks 8–11, 4 Weeks, 40 Hours)

Week	Focus Area	What You'll Master
Week 8	Redis & Caching	Cache layers, sessions, invalidation, rate limiting
Week 9	Global Performance	Rendering strategy, edge thinking, CDN-aware delivery
Week 10	Docker	Multi-stage builds, compose workflows, containerized delivery
Week 11	Observability	Metrics, dashboards, error tracking, structured logs

Phase 4: Cloud Delivery, Advanced AI & Career Prep (Weeks 12–14, 3 Weeks, 30 Hours)

Week	Focus Area	What You'll Master
Week 12	Cloud Delivery Foundations	Production deployment planning, environment separation, release discipline
Week 13	Advanced AI Systems	RAG-aware product thinking, tool usage, integration boundaries
Week 14	Career Launch	Architecture storytelling, resume, portfolio narrative, mock interviews

Skills You'll Gain

Languages & Frameworks

Technology	Proficiency Level
TypeScript (Advanced)	Production Architecture
Next.js 15 (Advanced)	Server Components, Edge, Performance
NestJS	Enterprise Backend Architecture
Turborepo	Monorepo Management
Redis	Caching, Sessions, Rate Limiting
RabbitMQ	Event-Driven Architecture
Docker	Containerized Delivery

Development Tools

Tool	Application
Playwright	End-to-end testing
Docker Compose	Multi-service orchestration
GitHub Actions	CI/CD pipelines
Grafana/Prometheus	Observability & monitoring

Tool	Application
Kubernetes (Introduction)	Container orchestration basics

Topic Depth and Awareness

Section	Guidance
Purpose	This course intentionally separates what you need to master in depth from what you only need to understand with working awareness.
Depth	The production engineering workflows, architecture decisions, and implementation patterns practiced repeatedly in class The execution areas you are expected to perform independently in advanced full-stack work The systems thinking most likely to matter in senior interviews, technical leadership, and real production delivery
Awareness	Adjacent tools, optional stretch topics, and industry context introduced for broader understanding Concepts you should be able to explain, compare, and recognize even if you are not yet executing them independently Advanced directions for later specialization, higher-level tracks, or guided self-study
How to use this syllabus	Spend most of your self-study time strengthening the depth topics first. Treat awareness topics as context builders that help you make better decisions and understand the larger professional landscape.

Project Pool

> *All options below are **intermediate-level final projects**. Each student chooses **one** final project from this pool. Trainers may run smaller guided exercises during the course, but public phase-wise project sections are intentionally removed so the completion standard stays clear and consistent.*

#	Final Project Choice	What You Will Build	Core Stack / Tools
1	Multi-Tenant Admin Platform	Build a secure admin product with tenant isolation, roles, billing boundaries, and operational polish.	Next.js 15, NestJS, PostgreSQL, Row-Level Security

#	Final Project Choice	What You Will Build	Core Stack / Tools
2	Realtime Collaboration Workspace	Build a shared workspace with live updates, robust state management, and conflict-aware UX.	React, WebSockets, TypeScript, real-time data
3	Subscription SaaS Backbone	Build the operational backbone of a SaaS product with subscriptions, queues, background jobs, and dashboards.	NestJS, RabbitMQ, Stripe, Turborepo
4	Observability-Ready API Platform	Build an API platform with caching, rate limits, tracing, observability dashboards, metrics, and failure visibility.	Redis, Edge / Node runtime, Prometheus, Grafana
5	Enterprise Workflow Automation Hub	Build a workflow-heavy business platform with approvals, async jobs, audit logs, and team visibility.	Next.js 15, queues, PostgreSQL, Sentry / observability

Career Paths & Trajectory

Role Path	Focus and Proof	Stage and Timeline	What Actually Matters
Full Stack Engineer II / Product Engineer	Own larger features, integration-heavy flows, and better system boundaries across a growing product. Proof you leave with: Enterprise deployment proof, observability habits, and architecture notes	Growth role - 2–4 years	Better trade-off decisions, stronger code review habits, and reliable delivery across the stack.
Senior Full Stack Engineer	Lead major feature areas, performance work, and architecture improvements with stronger production judgment. Proof you leave with: System design artifacts, debugging discipline, and cleaner infrastructure reasoning	Senior path - 4–6 years	Design systems, not just tickets, and explain long-term technical choices with evidence.
Tech Lead / Staff Engineer	Guide technical direction across services or product areas while helping other engineers raise quality. Proof you leave with: Architecture records, platform reasoning, and broader engineering influence	Leadership path - 6+ years	Judgment, mentorship, and keeping systems simpler, safer, and more maintainable under pressure.

Role Path	Focus and Proof	Stage and Timeline	What Actually Matters
Engineering Manager / Solution Architect	Move into people leadership or cross-system architecture after proving delivery, review quality, and long-term technical ownership. Proof you leave with: Mentoring patterns, incident leadership, and stronger engineering judgment	Alternative senior path - 7+ years	Scale decision quality, team effectiveness, and technical clarity - not just output volume.

Saarathi Gate & Completion Review

Before You Start: Saarathi Gate Assessment

All students complete the **Saarathi Gate Assessment** before Day 1. It is a short diagnostic review of aptitude, learning behaviour, and thinking style. It has **no pass/fail** and is used only to tailor support from the start.

After Course Completion: Saarathi Completion Review

The **Saarathi Academy Certificate** is issued after the selected final project is completed, documented, and reviewed by the trainer. There is **no separate certification exam** for this course.

Completion Requirements:

1. **Attendance:** Minimum 80% attendance
2. **Weekly Work:** Core deliverables, revision work, and practice tasks completed
3. **Final Project:** One intermediate-level project selected from the project pool and completed to trainer-approved quality
4. **Portfolio Proof:** Screenshots, documentation, case-study notes, or equivalent proof assets updated
5. **Trainer Review:** Practical execution, consistency, communication, and overall growth signed off by the trainer

Enrollment & Next Steps

Next Batch: Starting soon (contact for exact dates) **Offline Location:** Old Baneshwor Chowk, Kathmandu, Nepal **Mode:** Online + Offline **Contact (Call/WhatsApp):** 9761095364, 9744442469

» **[ENROLL NOW]** - Limited to 10 seats per batch

Senior engineers don't just build - they architect. In 14 weeks, you'll have the skills and the portfolio to prove it.

Last Updated: Mar 30, 2026