

TR-01-L1: Modern Full Stack Engineering (Foundation)

From Zero to Deployed - Your First Career in Tech Starts Here

Full-stack web development remains the most accessible and in-demand path into tech. This course helps disciplined entry-level learners with strong computer literacy become job-ready Junior Full Stack Developers, starting from the fundamental question "What is a computer?" and systematically building competency to create secure, deployed, production-grade web applications. In 16 weeks, you'll master JavaScript, React, Next.js, Node.js, PostgreSQL, and modern deployment - the exact stack powering companies from Kathmandu startups to Silicon Valley unicorns.

The focus is on logical thinking, syntax mastery, and industry-standard practices that prepare you for immediate employment in the global software engineering market.

Why This Course?

The Market Reality

Global Context: Full-stack developers who can move from UI to API to database remain one of the most practical entry points into software careers. Employers value builders who can ship complete features, collaborate with teams, and work comfortably across the modern web stack.

Nepal Context: Nepal's IT sector is projected to grow at 20%+ annually, with over 1,000 tech startups operating in Kathmandu Valley (Startups of Nepal, 2024). A strong full-stack foundation with React, Node.js, and PostgreSQL positions graduates for both local and global opportunities.

Your Opportunity: This course prepares you for **local tech jobs and global remote positions**. You'll graduate with deployed projects, a polished GitHub profile, and interview-ready skills that work anywhere.

| Nepal-Relevant Reality | Opportunity |
|---|--|
| Web development is still the fastest path into tech | Become job-ready with the right structure and projects |
| Nepal's tech ecosystem is growing rapidly | Startups, agencies, and IT companies need web builders |

| Nepal-Relevant Reality | Opportunity |
|--------------------------------------|---|
| Modern stacks change quickly | Learn current tools and deployment practices, not outdated ones |
| Remote work is accessible from Nepal | Build for the global market from Kathmandu |

Course Snapshot

| Parameter | Details |
|---------------|---|
| Course Code | TR-01-L1 |
| Title | Modern Full Stack Engineering (Foundation) |
| Duration | 4 Months (16 Weeks) |
| Schedule | Monday to Friday (Mon–Fri, 5 Days/Week), 2 Hours/Day |
| Total Hours | 160 Hours of Live Training |
| Batch Size | Maximum 10 Students |
| Course Fee | NPR 42,000 |
| Prerequisites | You should already be comfortable using a computer full-time, managing files and folders, and typing accurately. Complete freeCodeCamp Responsive Web Design (first 2 sections) before Day 1. Laptop with 8GB+ RAM. Saarathi Gate Assessment (diagnostic, no pass/fail) before Day 1. |
| Self-Study | Minimum 2 hours/day outside class (mandatory) |
| Outcome | Junior Full Stack Developer / Frontend Developer |

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 |

| Day | Activity |
|--------|---|
| 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: Programming Logic, Tools & Developer Foundations (Weeks 1–3, 3 Weeks, 30 Hours)

| Week | Focus Area | What You'll Master |
|--------|-----------------------|---|
| Week 1 | Developer Foundations | Computer thinking, terminal basics, VS Code, Git, GitHub, first scripts |
| Week 2 | JavaScript Part 1 | Variables, conditions, loops, functions, debugging habits |
| Week 3 | JavaScript Part 2 | Arrays, objects, async thinking, problem-solving patterns |

Phase 2: Frontend Engineering (Weeks 4–7, 4 Weeks, 40 Hours)

| Week | Focus Area | What You'll Master |
|--------|-----------------------------|---|
| Week 4 | HTML, CSS & DOM | Semantic HTML, layout, responsive basics, DOM interaction |
| Week 5 | Tailwind CSS | Utility workflow, component patterns, spacing, state styles |
| Week 6 | React Fundamentals | Components, props, state, events, forms |
| Week 7 | Advanced React & TypeScript | Hook patterns, TypeScript basics, routing, shared state |

Phase 3: Backend Engineering (Weeks 8–10, 3 Weeks, 30 Hours)

| Week | Focus Area | What You'll Master |
|--------|-------------------|---|
| Week 8 | Node.js & Express | Server basics, routing, middleware, request and response flow |
| Week 9 | PostgreSQL & SQL | Schema design, joins, filtering, aggregation, query thinking |

| Week | Focus Area | What You'll Master |
|---------|--------------------|---|
| Week 10 | API Best Practices | Validation, errors, uploads, security basics, documentation |

Phase 4: Full-Stack Integration, Auth, Testing, Deployment & Career Prep (Weeks 11–16, 6 Weeks, 60 Hours)

| Week | Focus Area | What You'll Master |
|---------|--------------------------|---|
| Week 11 | Drizzle ORM | Type-safe queries, relationships, migrations |
| Week 12 | Authentication | Sessions or tokens, auth flows, access control basics |
| Week 13 | Next.js Integration | App Router thinking, server and client boundaries, full-stack wiring |
| Week 14 | Testing & Code Quality | Unit, UI, and API tests; linting and review habits |
| Week 15 | AI Features & Deployment | Light AI-assisted features, deployment flow, environment awareness |
| Week 16 | Career Launch | Portfolio cleanup, project presentation, resume, LinkedIn, interview drills |

Skills You'll Gain

Languages & Frameworks

| Technology | Proficiency Level |
|--------------------------------|---|
| JavaScript (ES6+) & TypeScript | Professional |
| React 19 & Next.js 15 | Production-Ready |
| Node.js & Express | Server-Side Competent |
| PostgreSQL | Relational database design, queries, optimization |
| Drizzle ORM | Type-safe database access, migrations |
| REST APIs | Design, validation, documentation, security |
| Zod | Input validation & type safety |

Development Tools

| Tool | Application |
|-----------------------|---------------------------------------|
| Git & GitHub | Version control, collaboration, CI/CD |
| VS Code | Professional development environment |
| Tailwind CSS v4 | Utility-first UI styling |
| Vercel AI SDK | AI feature integration |
| Docker (Introduction) | Container 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 | <p>The core tools, coding workflows, and portfolio projects practiced repeatedly in class</p> <p>The execution areas you are expected to perform independently by the end of the course</p> <p>The concepts most likely to appear in assessments, project reviews, interviews, and entry-level work</p> |
| Awareness | <p>Adjacent tools, optional stretch topics, and industry context introduced for broader understanding</p> <p>Concepts you should be able to explain, compare, and recognize even if you are not yet executing them independently</p> <p>Advanced directions for later specialization, higher-level tracks, or guided self-study</p> |
| 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 | Task Management SaaS | Build a complete task manager with auth, CRUD flows, roles, and a clean responsive UI. | Next.js 15, React, TypeScript, PostgreSQL |
| 2 | E-Commerce Catalog App | Build a product catalog and cart system with filtering, checkout flow, and clear data handling. | Next.js 15, Tailwind CSS, database modeling, form validation |
| 3 | Portfolio CMS Platform | Build a content-managed portfolio site with admin editing, media, and deploy-ready structure. | Next.js 15, server actions, auth, file handling |
| 4 | Community Feed App | Build a small social or discussion platform with posts, comments, and moderation basics. | React, API routes, PostgreSQL, authentication |
| 5 | Booking & CRM System | Build a booking workflow with customer records, status updates, reminders, and dashboard views. | Next.js 15, forms, role-aware UI, database queries |

Career Paths & Trajectory

| Role Path | Focus and Proof | Stage and Timeline | What Actually Matters |
|---|--|--------------------------------|---|
| Junior Frontend Developer / Junior Full Stack Developer | Support feature delivery across UI, forms, API integration, and simple database-backed workflows. Proof you leave with: Deployed projects, CRUD flows, and end-to-end feature proof | Entry role - first 0–12 months | Reliable execution, clean Git habits, and being able to explain your own feature work clearly. |
| Full Stack Developer | Build and maintain complete product features across frontend and backend layers with less supervision. Proof you leave with: Production-style UI work, API delivery, and stronger debugging confidence | Growth role - 1–3 years | Own a feature from ticket to deploy, communicate blockers early, and write code teammates can extend. |

| Role Path | Focus and Proof | Stage and Timeline | What Actually Matters |
|--|--|-----------------------------|--|
| Backend-Focused Full Stack Engineer / Product Engineer | Take stronger ownership of APIs, data models, auth flows, and technical trade-offs inside a product team. Proof you leave with: REST patterns, database work, and better system reasoning | Specialist path - 3–5 years | Better judgment around architecture, reliability, and simplifying work before adding complexity. |
| Senior Full Stack Engineer / Freelance Product Developer | Lead feature areas or ship client-facing products with stronger ownership of delivery, scope, and quality. Proof you leave with: Portfolio proof, deployment habits, and clearer product execution | Senior path - 4+ years | Consistent delivery, calm debugging, and the ability to make sensible trade-offs without drama. |

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:

- Attendance:** Minimum 80% attendance
- Weekly Work:** Core deliverables, revision work, and practice tasks completed
- Final Project:** One intermediate-level project selected from the project pool and completed to trainer-approved quality
- Portfolio Proof:** Screenshots, documentation, case-study notes, or equivalent proof assets updated
- 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

The demand for full-stack developers isn't slowing down. In 16 weeks, you'll have the skills and the portfolio to prove it.

Last Updated: Mar 30, 2026

