

— TR-01

Full Stack Development with Vibe Coding

COURSE BRIEF

SKILLS & TOOLS YOU WILL MASTER



Cursor + Claude Code

NEXT:

Next.js + Supabase



Flutter + Dart



v0 + Lovable



Vercel + Domains

10 Reasons This Works.

Why students choose Saarathi – and how each card helps you ship a real career.

01

10 Max / Batch

Not 50. Not 30. Ten. Every student seen, heard, and unblocked – every class.

02

Sunday Open Classroom

Every Sunday the classroom is open. Come in, pair-program, get unstuck, or just focus.

03

Weekly Detailed Syllabus

Full syllabus from day one. Every week: what to read, what to build, what the outcome is.

04

Revision Every 5th Day

Every fifth class is a revision session. Nothing piles up or slips through.

05

Week 1: Mindset

The first week is about getting into the right headspace before touching a line of code.

06

3 Ways of Learning

Self-learning + collaboration + mentorship. All three built into every course.

07

1:1 Mock Interview

Real mock interview with your mentor in the final week. Actual hiring prep, not fluff.

08

CV + LinkedIn

Resume writing and LinkedIn optimisation. Leave with a profile that gets replies.

09

Real Projects

Every course ends with a deployed, publicly accessible project you can show employers.

10

Gate Assessment

Aptitude test before Day 1. Maps your strengths, weaknesses, learning style, and pace – then builds your personal plan.

saarathiacademy.com.np/full-stack-development-with-vibe-coding-course-in-nepal

+977-9744442469 · +977-9761095364

Full Stack Development with Vibe Coding

Why This Course

The way software gets built has changed. With AI tools like Cursor, Claude Code, GitHub Copilot, v0, Lovable and Bolt, one person can describe an idea and direct AI to build a real website, web app or mobile app in days, not months. This way of working is called "vibe coding": you stay in flow, describe what you want, and the AI writes the code. Don't worry if none of that means anything yet, we start from zero and the first two weeks teach how technology actually works before you build a thing.

But vibe coding done badly is dangerous: you paste code you don't understand, it breaks, and you are stuck. AI gets you the first 70% of an app in minutes, but the last 30% (the bugs, the security, the edge cases) is where people who can't read code get trapped, and audits of AI-built apps keep finding the same avoidable flaws, like an exposed key or a login that does not actually protect anything. The skill this course builds is vibe coding done right, directing AI well and then reading, verifying, fixing and deploying what it writes, so you own that last 30%. That is exactly what this six-week skill module teaches, and it is the difference between our graduates and someone who only knows how to prompt.

Six weeks is enough time to build the base properly and still ship real work. A full third of the course, the first two weeks, is pure foundation, then it rests on one decision that removes a whole month of busywork: **Next.js is the entire stack**. With the App Router, Server Components, Server Actions and Route Handlers, one Next.js app is your frontend and your backend at once, so there is no separate backend server to learn, no second language and no second deployment. You build one app, and the Flutter mobile app in Week 6 calls that same app's Route Handlers. That is how AI-assisted teams actually ship in 2026, and it is why a complete beginner can ship two real, deployed builds in six weeks: a full-stack web app on a real custom domain, and a Flutter mobile app running on their own Android phone.

Who Is This Course For

- **Complete beginners** who have never written a line of code and want to ship real apps fast
- **Founders and aspiring founders** who want to build their own product without hiring a whole team
- **Students and recent graduates** who want concrete, deployed projects, not just tutorials
- **Career changers** from any background who want a fast, practical start in tech
- **Designers and product people** who want to build and ship their own ideas end to end
- **Freelancers** who want to deliver complete website and app work with AI

- **Anyone who wants one focused, intensive month and a half** rather than a course stretched over a season

Course Snapshot

PARAMETER	DETAILS
Course Code	TR-01
Duration	6 Weeks (1.5 Months)
Schedule	Monday to Friday, 2 Hours/Day
Total Hours	60 Hours
Batch Size	Maximum 10 Students
Course Fee	NPR 15,500
Level	Beginner → Full Stack Builder (Web & Mobile)
Outcome	Build and ship a real full-stack web app and a real mobile app with AI, and deploy them live to a real domain, with the first two weeks spent building your foundation

Prerequisites

- No coding experience required, you start from zero
- A laptop with 8GB+ RAM (macOS, Windows or Linux) and an Android phone for testing your mobile build
- Comfortable using a computer daily, files and folders
- Basic English reading, docs and error messages are in English
- Plan 2 hours of self-study after every class, that is where the learning lands
- Saarathi Gate Assessment before Day 1, diagnostic, no pass or fail

Nothing to install beforehand. VS Code, GitHub, Vercel, Cursor and Claude are all free and we set them up together in the first class.

One thing worth starting early: apply for your free .com.np domain at register.com.np. It needs a citizenship scan and approval takes a few days, so start it in Week 1 and it is ready for your real-domain deploy in Week 6.

Assessment

The **Saarathi Gate Assessment** runs before Day 1 as a diagnostic with no pass or fail; it maps your learning style and pace so trainers can tune support and pacing from day one.

Your Learning Week

DAY	ACTIVITY
Mon–Fri	2-hour live class session (hands-on, build-driven) with mentor
Mon–Fri	2-hour self-study at home (building, prompting, reading and fixing AI code)
Sunday	Open classroom for focused build time or rest
Day 5 (every week)	Within the 2-hour session, the last 1 hour is a built-in revision block for consolidation, doubt-clearing and a preview of the week ahead

Real improvement happens when you build, break, debug and re-ship between classes. The classroom gives structure; the hours between classes give depth. The self-study hours are not optional extra credit, they are half the course.

Week-by-Week Curriculum

Phase 1 · Foundations: How Tech Works, Concepts, Whys & Hows (Weeks 1–2, 20 Hours)

WEEK	FOCUS AREA	TOPICS
Week 1	How the Web & Software Actually Work	– How the internet works and why: clients, servers, browsers, requests and responses – What really happens when you open a URL: what DNS does as the internet's phonebook, what an HTTP/HTTPS request and response carry, and what the padlock means – What a web app is and what its parts do (frontend, backend, database, deploy) – Why one Next.js app can be both the frontend and the backend, through the App Router, Server Components, Server Actions and Route Handlers, so there is no separate backend server – What a web server and a reverse proxy (Nginx) do, and why a host like Vercel handles that layer for you – What programming actually is, as a concept: instructions, logic

WEEK	FOCUS AREA	TOPICS
		and data, without writing syntax; plus core tech vocabulary as clear mental models, not jargon
Week 2	Data, APIs, Security & the AI Mindset	- Databases big picture: relational (PostgreSQL) vs NoSQL (MongoDB), tables and relationships vs documents, when and why each - What an ORM (e.g. Prisma) is and why it sits between your code and the database - What an API is and how systems talk (REST), and why it matters for both the web app and the Week 6 mobile app - What state and state management mean and why; and caching with Redis: why apps cache, sessions and speed - What version control and GitHub are for; why security, secrets and environments matter, so the most common AI-app mistake (an exposed key) is one you understand before you can make it - Why AI changes who can build software, why it gets you the first 70% fast but the last 30% (bugs, security) needs someone who can read the code, what vibe coding really means, and how to direct AI well: describe, read, verify, fix

Phase 2 · Your Workshop, Git & Your First Live Next.js App (Week 3, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 3	Your Workshop, Git & Your First Next.js Build	- Set up your workshop: install VS Code, sign in to Cursor and Claude, and confirm everything works - The editor, the terminal and how files and folders are organised in a project, in plain words - What Git and GitHub really do, hands-on: make your first repository, save a commit, and push to GitHub - Meet the AI build tools: how Cursor, Claude Code and GitHub Copilot fit into building, where the no-code and low-code generators (v0, Lovable, Bolt) fit, and how to write a clear instruction - How to read AI-generated code and follow what it does, line by line, the core skill the whole course rests on - Why this stack: what problem Next.js, React, TypeScript and Tailwind CSS each solve, then scaffolding (generating the skeleton of) that Next.js (App Router) + React + TypeScript + Tailwind CSS app with AI, reading its folder structure and the split between server and client code

WEEK	FOCUS AREA	TOPICS
		- First live URL on Vercel: something you directed AI to build is on the internet in Week 3

Phase 3 · Full-Stack Core: Database, Secrets, Auth & Roles (Week 4, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 4	Full-Stack Core: Database, Secrets, Auth & Roles	- A real database: how relational data works, then Supabase / PostgreSQL wired up through an ORM (Prisma), with Firestore / MongoDB as the NoSQL alternative and where Redis caching fits as the app grows - Environment variables and secrets across local and production: the Supabase publishable key (safe in the browser) vs the secret key (never), kept out of your code and off GitHub, closing the most common AI-app mistake - How login actually works: session tokens in plain words, then real signup and login with Supabase auth, and protected pages - Roles and permissions so not every logged-in user can do everything, checked on the server - Reading real data onto the page for the signed-in user (the read of CRUD) and holding state in the UI (logged-in user, lists), so each user logs in and sees their own data - A real auth-and-database foundation, verified against the live database

Phase 4 · Full-Stack for Real: CRUD, Forms, APIs & Debugging (Week 5, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 5	Full-Stack for Real: CRUD, Forms, APIs & Debugging	- Full CRUD on real data (create, update, delete on top of last week's read) built with AI-assisted components, with UI state kept correct - Forms with validation (catching bad input on client and server) - File and image upload with public vs private buckets, served back safely - Search and filter over real data, verified against the live database - Server Actions vs Route Handlers: your backend, in the same app, no separate server; and why a Route Handler serving clean REST is what your Flutter app calls in Week 6

WEEK	FOCUS AREA	TOPICS
		- A real integration with an outside service (payments in test mode, bookings, email, maps or an AI API), with errors handled so it fails safely, not silently - Debugging when AI gets it wrong (reproduce, read, hypothesise, fix, verify), then ship the full-stack web app, published and shareable

Phase 5 · Flutter Mobile App & Deploy for Real (Week 6, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 6	Flutter Mobile App & Deploy for Real	- Why cross-platform: one codebase becomes both an Android and an iOS app; Flutter (Google's mobile-app framework) and Dart (its language) with AI assist, plus FlutterFlow (visual screens, real Flutter code) - Screens, navigation (moving between screens) and state (live data like the logged-in user) - One backend, two apps: the mobile app calling the same Route Handlers and the same Supabase / PostgreSQL data as your web app, with no second server - Reading and fixing AI-generated Flutter code, then building and running the app on a real Android phone - Play Store / App Store basics: what publishing involves (awareness) - An advanced AI feature via an LLM API (in-app chat, semantic search or AI summaries), with reusable components and a cleaner, maintainable structure; webhooks and scheduled jobs at awareness level - Production hardening and a security pass on secrets and access control (the two flaws AI-built apps fail on most) - Deploy to a real custom domain with HTTPS on Vercel, with basic monitoring (know when it breaks), analytics (see real usage), performance and SEO basics, then debugging the issues that only show up in production - Both builds live and stable on a real domain

Skills You Will Gain

CATEGORY	TECHNOLOGIES
AI Build Tools	Cursor, Claude Code, GitHub Copilot, v0, Lovable, Bolt, prompt engineering
Web	Next.js (App Router, Server Components, Server Actions, Route Handlers), React, TypeScript, Tailwind CSS, state management
Backend & Data (inside Next.js, no separate server)	Supabase, PostgreSQL, MongoDB, Prisma (ORM), Redis (caching), Firebase, REST APIs, authentication, roles and permissions
Infrastructure	Web servers, reverse proxy (Nginx), hosting (concepts)
Mobile	Flutter, Dart, FlutterFlow
Deployment	Vercel, custom domains, Play Store / App Store basics, monitoring, analytics
Core Skill	Reading, verifying, debugging and fixing AI-generated code
Tools	Git, GitHub, VS Code

Topic Depth and Awareness

Depth means repeated practice until students can do it confidently. **Awareness** means students are introduced clearly so they can read, discuss or learn it independently later.

FOCUS AREA	LEARNING FOCUS
Vibe-coding workflow: prompt, read, verify, fix	Depth
Building websites and web apps with AI	Depth
Next.js (App Router, Server Components, Server Actions, Route Handlers), React, TypeScript, Tailwind CSS	Depth
Supabase / PostgreSQL, Prisma, auth, roles, real data	Depth

FOCUS AREA	LEARNING FOCUS
Modern-stack concepts: PostgreSQL, MongoDB, ORMs (Prisma), state management, Redis caching, Nginx, APIs	Depth (concepts; AI writes the code)
Flutter and Dart mobile apps with AI	Depth
Deployment: domains, Vercel, app stores	Depth
Monitoring and analytics basics	Awareness
Automated testing	Awareness (introduced, not drilled in a six-week module)
Logic and problem-solving (freeCodeCamp JavaScript drills in weekly revision hours)	Awareness
Deep computer-science fundamentals and algorithms	Awareness (covered in dedicated tracks)
Separate backend services (Node.js, Express, standalone API servers)	Awareness (not needed here, Next.js is the backend; covered in depth in TR-12)
Production infrastructure (Docker, CI/CD at scale: the tooling teams use to package and auto-release software)	Awareness (covered in depth in TR-03)

FAQ

What does this course cost and how do I pay?

NPR 15,500 for the full 6-week (1.5-month) program. Classes run Monday to Friday for 2 hours per day, giving you 60 guided hours plus the practice expected outside class. Pay by cash at our Old Baneshwor campus, bank transfer, or Fonepay. Email info@saarathiacademy.com.np for the payment plan.

Do I need prior experience?

No. You start from zero, and the first two weeks are a full foundation before you build anything. The Saarathi Gate Assessment runs before Day 1 to map your starting point, diagnostic, not pass-or-fail.

Can a complete beginner really do this in six weeks?

Yes, and the pacing is built for it. A full third of the course, the first two weeks, is spent building your base, so nothing later is a mystery. AI writes the code, so your job is to direct, read, verify and fix rather than memorise syntax. And Next.js is the entire stack, so there is no separate backend server, no second language and no second deployment to learn. What is left is 60 guided hours plus 2 hours of daily self-study, and the self-study is not optional.

What is "vibe coding" and will I actually learn?

Vibe coding means you describe what you want and direct AI tools to build it. AI gets you the first 70% of an app fast, but the last 30% (the bugs, the security, the edge cases) is where people who only prompt get stuck. We teach the disciplined version: you read, verify and fix the code the AI writes, so you understand and own what you ship instead of pasting blindly. That is the difference between our graduates and someone who only knows how to type a prompt.

Why is there no separate backend in this course?

Because you do not need one. A single Next.js app with the App Router, Server Components, Server Actions and Route Handlers is your frontend and your backend at once. Your database work, your login, your roles and your REST endpoints all live in that one app, and in Week 6 the Flutter mobile app calls those same Route Handlers. You still learn the concepts of the wider stack (MongoDB, Redis, reverse proxies like Nginx) so you understand what a bigger system looks like. If you later want to build standalone API services with Node.js and Express, that is TR-12.

Is this more about websites and web apps, or mobile?

Web first. Five of the six weeks are foundations, your first Next.js build, and a full-stack web app with a real database, login, roles and a real integration; Week 6 is a Flutter mobile app on your own phone talking to that same backend, plus deploying everything live on a real domain. APIs, monitoring and deployment are taught along the way.

What will I build by the end?

Two real builds, both deployed: a full-stack web app with login, roles and real data on a real custom domain, and a Flutter mobile app running on your own Android phone that talks to the same backend. Plus an advanced AI feature, and a GitHub repo you can keep building on.

How does the batch and mentorship work?

Maximum 10 students per batch, every student gets direct mentor time and project review. Hybrid by default, in-person at Old Baneshwor with online recordings and office hours. Sunday Open Classroom is free for enrolled students.

Enrollment

- **Next Batch:** see the live course page for the current slot
 - **Location:** Old Baneshwor Chowk, Kathmandu, Nepal
 - **Mode:** Online + Offline
 - **Contact:** 9744442469, 9761095364
 - Limited to 10 seats per batch
-

Full Stack Development with Vibe Coding — full details, fees and next batch dates:
saarathiacademy.com.np/full-stack-development-with-vibe-coding-course-in-nepal

Version 1.0 · Updated July 18, 2026