

Full Stack Development with Vibe Coding

Duration

6 weeks

Live classes

60 hours

Schedule

Mon to Fri, 2 hrs/day

Batch size

10 students max

Tools you will use

 Cursor + Claude Code Next.js + Supabase Flutter + Dart v0 + Lovable Vercel + Domains

10 students a batch. Exceptional mentors who guide you personally, and a **syllabus built from real job posts**, cuz we never skip the "hard parts" 😊

The class

10 per batch**Ten students. Never more.**

A hard cap, not a target. Your mentor knows your name and your work.

The mentor

4–5 yrs in the job**Someone who does the job.**

Working professionals, not full-time teachers. They read your work line by line.

The syllabus

1/3 is fundamentals**The hard parts, not just the basics.**

Teaches everything the job needs and never skips the hard parts most courses skip, built from real job descriptions. Check the syllabus before you enroll.

Built into every course

Gate assessment

A short test before Day 1 finds your strengths and pace, then sets your plan.

A real, live project

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

Revision every 5th class

Every fifth class goes back over the week, so nothing piles up.

1:1 mock interview

A real interview practice with your mentor in the final week.

Certification exam

Pass the exam at the end of the course to earn your certificate.

CV and LinkedIn help

We help you rewrite your CV and LinkedIn before you apply.

Full Stack Development with Vibe Coding

Why This Course

With AI tools like Cursor, Claude Code and Copilot, one person can now build a real app in days. Building apps by describing them to AI is called "vibe coding". We teach the version that actually works in real life.

- **You learn the basics first.** The first two weeks teach how technology works, before you build anything.
- **You learn the part AI gets wrong.** AI writes about 70% of an app, but the last 30% (bugs, security and unusual cases) traps people who cannot read code.
- **You check everything the AI writes.** You read it, test it, fix it and then put it online.
- **You learn one tool for the whole app.** Next.js builds both the frontend (what users see) and the backend (the server side), so you do not need a second language.
- **You build two real apps.** One is a web app on your own domain, and one is a Flutter mobile app on your phone.
- **You learn to get paid.** The last two weeks show you how to turn these builds into your first paid job.

Who Is This Course For

- **Complete beginners** who have never written code.
- **Founders** who want to build their own product without hiring a team.
- **Students and graduates** who want real projects running online, not just tutorial exercises.
- **Career changers** who want a practical start in tech.
- **Designers and product people** who want to build and launch their own ideas.
- **Freelancers** who want to deliver complete websites and apps with the help of AI.

Course Snapshot

PARAMETER	DETAILS
Course Code	TR-01
Duration	6 Weeks (1.5 Months)
Schedule	Monday to Friday, 2 Hours/Day

PARAMETER	DETAILS
Total Hours	60 Hours
Batch Size	Maximum 10 Students
Course Fee	NPR 21,000
Level	Beginner → Full Stack Builder (Web & Mobile)
Outcome	Build, secure and ship a real full-stack web app and a real mobile app with AI, deploy them live to a real domain, and learn to turn that work into your first paid gig

Prerequisites

- You do not need any coding experience. You start from zero.
- You need a laptop with at least 8GB of RAM (macOS, Windows or Linux) and an Android phone for testing.
- You should be comfortable using a computer, files and folders.
- You need basic English reading, because guides and error messages are in English.
- You need time for 2 hours of self-study after every class.
- You take the Saarathi Gate Assessment before Day 1. It is not a pass or fail test.

You do not need to install anything beforehand. VS Code, GitHub, Vercel, Cursor and Claude are free, and we set them up together in the first class.

Apply early for your free .com.np website address at register.com.np in Week 1. It needs a scan of your citizenship and takes a few days. That way it is ready when you put your app online in Week 5.

Assessment

- **Gate Assessment.** You take it before Day 1. It is not pass or fail. It shows us how you learn and how fast.

Projects

Your mentor gives you 1 project during the course. It needs to be approved before you get your certificate and before the Career Launchpad opens.

Your Learning Week

DAY	ACTIVITY
Mon–Fri	2-hour live class with your mentor, where you build real apps
Mon–Fri	2 hours of self-study at home, where you build, give AI instructions, and read and fix its code
Sunday	Open classroom for extra build time, or a day of rest
Day 5 (every week)	The last hour of class is for revision, clearing doubts and a preview of next week

Class gives structure. Practice between classes gives depth. Self-study is half the course, so it is not optional.

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 and data, without writing syntax; plus core tech vocabulary as clear mental models, not jargon

WEEK	FOCUS AREA	TOPICS
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 4 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 Live Next.js App	- 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 - First live URL on Vercel: something you directed AI to build is on the internet in Week 3

Phase 3 · Full-Stack Build, Security & Testing (Week 4, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 4	Full-Stack Build, Security & Testing	- A real database: how relational data works, then Supabase / PostgreSQL wired up through an ORM (Prisma), with Firestore / MongoDB as the NoSQL alternative and Redis caching, plus environment variables and secrets kept out of your code and off GitHub - How login actually works, then real signup and login with Supabase auth, protected pages, and roles and permissions checked on the server - Full CRUD, forms with validation, file and image upload, search and filter, Server Actions vs Route Handlers serving clean REST, and one real integration (payments, bookings, email, maps or an LLM API feature like semantic search), with reusable components, and webhooks and scheduled jobs at awareness level - Flutter (Google's mobile-app framework) and Dart with AI assist plus FlutterFlow: screens, navigation and state. One backend, two apps: it calls the same Route Handlers and data as the web app, with no second server, and is built and run on a real Android phone - Reading and fixing AI-written code, deepened: reading a diff, tracing a bug to its source, asking the AI a sharper follow-up question - A security checklist every build must pass: no secret exposed, every protected page checked on the server, every input validated, database access rules enforced - Testing by hand: a written checklist, every path clicked, the app broken on purpose

Phase 4 · Ship and Maintain a Live App (Week 5, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 5	Ship and Maintain a Live App	- The full-stack web app deployed live to a real custom domain with HTTPS on Vercel - Basic error monitoring, so you find out an app broke from a tool, not an angry user - Usage analytics, so you can see how the app is actually used - Performance and SEO basics: loading speed, image sizing, mobile layout, page titles - Fixing a real bug the way it happens after launch: a report comes in, you reproduce it, read the error and the code, fix it, verify it live

WEEK	FOCUS AREA	TOPICS
		- Updating a live app without breaking it: test first, deploy safely, confirm nothing else broke

Phase 5 · The Client Side of the Job (Week 6, 10 Hours)

WEEK	FOCUS AREA	TOPICS
Week 6	The Client Side of the Job	- Turning a vague request ("I want an app for my shop") into a written scope: what it does, what it does not, and by when - Quoting and estimating the work like a freelancer or an agency prices a small build in Nepal - Writing a client brief a non-technical reader can approve before any code is written - Handover docs a client can use without you, plus a recorded walkthrough video, the way real freelance work is actually handed off - A timed live build under real time pressure, AI allowed, every choice explained out loud - Portfolio: presenting both builds the way a client or interviewer would actually look at them - A mock interview or client call, with mentor feedback, aimed at your first paid gig or your first job

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

CATEGORY	TECHNOLOGIES
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 you practise it until you can do it confidently. **Awareness** means we introduce it clearly so you can learn more 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
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

FOCUS AREA	LEARNING FOCUS
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 21,000 for the full 6 weeks, 60 guided hours, Monday to Friday, 2 hours a day. 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 with two full weeks of basics. The Gate Assessment before Day 1 is not a pass or fail test.

Can a complete beginner really do this in six weeks?

Yes. The first two weeks build your base. After that, AI writes the code and Next.js keeps everything in one app. Your job is to guide the AI, read its code, check it and fix it, with daily self-study.

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

In vibe coding, you describe what you want and AI builds the first 70%. The last 30% (bugs, security and unusual cases) traps people who only give instructions. So we teach you to read, check and fix the code yourself.

Why is there no separate backend in this course?

You do not need one. One Next.js app works as both your frontend and your backend, and the Flutter app talks to it directly. You still learn what MongoDB, Redis and Nginx are for. Building separate backend servers with Node.js and Express is taught in TR-12.

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

It is web first, with a mobile app included:

- **Weeks 1 to 3:** You learn the basics and build your first Next.js app.
- **Week 4:** You build a full web app and a Flutter app on your phone.
- **Week 5:** You put the web app online on your own domain.
- **Week 6:** You learn to plan and price client work, hand it over, and practise an interview.

What will I build by the end?

- A complete web app with login, user roles and real data, running on your own domain.
- A Flutter app on your Android phone that uses the same backend.
- An advanced AI feature, plus a GitHub repository (your online code folder) to keep building on.

How does the batch and mentorship work?

- Each batch has at most 10 students, and each student gets direct time with the mentor and project reviews.
- Classes are in person at Old Baneshwor. Recordings and online office hours are also available.
- The Sunday Open Classroom is free for enrolled students.

Will this get me paid work or a job?

The last two weeks are built for that. You finish with:

- A live app, a written plan and price for a client, and a proper handover.
- A timed build and a practice interview.

Most graduates freelance first, then move to a junior job or an agency. Six weeks will not make you a senior developer, but it can get you your first paid job.

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 2.0 · Updated 11 Ashwin 2083

