Glossary: Vibe Coding Terms

> *Your non-jargon guide to technical terms. No condescension, just clarity.*

A

API (Application Programming Interface) The waiter in the restaurant analogy. It's the communication system that carries messages between different parts of your app - like taking an order from the front-end to the kitchen (back-end), and bringing back the results.

Example: When you click "Save Post" on Instagram, an API carries that post to Instagram's servers and brings back a confirmation.

Authentication (Auth) The bouncer at the door of your app. It's the system that lets users create accounts, prove who they are (log in), and stay recognized as they use your app.

Example: When you log into Netflix and it remembers you're watching Season 2 of that show - that's authentication working.

Avatar A profile picture or image that represents a user. Usually small, square, and shows up next to their name throughout the app.

Example: The circular photo next to your name in Gmail.

B

Back-End The kitchen in the restaurant analogy. It's where the real work happens behind the scenes - processing data, making decisions, talking to the database, and handling all the logic users never see.

Example: When you submit a login form, the back-end checks if your password is correct and decides whether to let you in.

bcrypt A security tool that scrambles passwords into unreadable text before storing them. You'll never touch it directly - AI handles this automatically when you say "users should have secure login."

Example: It turns "MyPassword123" into something like "$2a$10$N9qo8uLOickgx2ZMRZoMyeIjZAgcfl7p92ldGxad68LJZdL17lhWy"

Blob Storage / Object Storage A digital warehouse where files (images, PDFs, videos) are stored. Different from a database - this is specifically for files, not structured data like names or numbers.

Example: All the photos you upload to your app live in blob storage, while your account name lives in the database.

Build The process of converting your human-readable code into optimized files that browsers and servers can run super fast. Like translating a recipe into restaurant kitchen shorthand.

Example: Your Next.js code gets "built" into static files and optimized JavaScript before going live.

Bug When something in your app doesn't work the way you intended. Could be a button that doesn't click, data that doesn't save, or any unexpected behavior.

Example: "There's a bug where the profile picture won't upload" means that feature is broken.

C

CDN (Content Delivery Network) A network of servers around the world that store copies of your website. When someone visits from Tokyo, they get the version from a Tokyo server instead of waiting for data from a US server.

Example: Like having franchise locations of your restaurant in every major city instead of making everyone fly to the original location.

Claude An AI assistant made by Anthropic that can understand your requests, write code, help you debug, and explain technical concepts. Think of it as your co-developer who never sleeps.

Example: You tell Claude "I need users to be able to save bookmarks" and it generates all the code to make that happen.

Column A single category of information in a database table. Think of it like a column in a spreadsheet - one column for names, another for emails, another for signup dates.

Example: In a "users" table, you might have columns for email, password, name, and signup_date.

CORS (Cross-Origin Resource Sharing) Security rules that control which websites can talk to your app's back-end. AI handles this automatically - you just need to know it exists in case you see the error.

Example: Without CORS setup, your app at myapp.com couldn't request data from api.myapp.com.

Cursor An AI-powered code editor that lets you write natural language requests and generates code across your entire project. Like VS Code with a brain.

Example: You can highlight code and type "make this mobile-friendly" and Cursor rewrites it.

CSS (Cascading Style Sheets) The styling language that makes websites look good - colors, fonts, layouts, animations. With Tailwind (see below), you rarely write CSS directly.

Example: CSS is what makes a button blue with rounded corners instead of a boring gray rectangle.

D

Database The pantry in the restaurant analogy. It's where all your app's data lives permanently - user accounts, posts, comments, settings, everything that needs to be remembered.

Example: When you save a bookmark, it goes into a database so it's still there when you come back tomorrow.

Deploy / Deployment The act of taking your app from your computer and putting it live on the internet where other people can actually use it. Like opening your restaurant to the public.

Example: "I deployed to Vercel" means your app is now live at a real URL that anyone can visit.

Domain Your app's address on the internet, like myapp.com. You buy these from domain registrars like Namecheap for about $12/year.

Example: "buildtolaunch.com" is the domain for Jenny's newsletter.

Drip Campaign A series of automated emails sent over time, usually for onboarding or nurturing users. Day 1: welcome. Day 3: tips. Day 7: case study.

Example: When you sign up for a newsletter and receive a carefully timed series of emails introducing the topic.

E

Environment Variable Secret information your app needs but shouldn't be visible in code - like API keys, database passwords, or Stripe secret keys. Stored separately and securely.

Example: Your Stripe secret key is stored as an environment variable so it's not visible in your code on GitHub.

Edge Function Code that runs on servers close to your users geographically, making your app faster. Vercel and other platforms handle this automatically.

Example: A user in Australia gets their data from a server in Sydney, not from a server in California.

F

Flask A lightweight Python framework for building web applications. Popular for data-heavy apps because Python has powerful data processing libraries.

Example: If you're building a data analysis tool and already know Python, Flask is a good choice for the back-end.

Foreign Key A reference in one database table that points to data in another table. This is how you create relationships between data.

Example: In a "posts" table, the user_id column is a foreign key that points to the user who wrote that post.

Framework A pre-built foundation for building apps. Instead of building everything from scratch, you build on top of a framework that already solves common problems.

Example: Next.js is a framework that gives you routing, optimization, and deployment tools out of the box.

Front-End The dining room in the restaurant analogy. It's everything users see and interact with - buttons, forms, images, text, navigation. The visual part of your app.

Example: The login screen with email and password fields is the front-end. What happens when you click submit is the back-end.

G

Git A version control system that tracks every change you make to your code. Like "Track Changes" in Microsoft Word but for code. Lets you undo anything and collaborate with others.

Example: You make changes, use git to save a snapshot with a message like "Added login feature", and can always go back to that version.

GitHub A website where you store your code online using Git. Think of it as Google Drive for code, with collaboration features.

Example: You push your code to GitHub, and then Vercel automatically deploys it whenever you update it.

Google OAuth The "Sign in with Google" button. OAuth is the secure process that lets users log into your app using their Google account without you ever seeing their Google password.

Example: "Sign in with Google" where you click, approve access, and you're logged in - that's OAuth.

H

Hash / Hashing A security process that turns passwords into scrambled text that can't be unscrambled. One-way encryption. You never need to implement this - AI does it automatically.

Example: Your password "hello123" gets hashed to "8b1a9953c4611296a827abf8c47804d7" before being stored.

Hosting Platform A service that keeps your app running on the internet 24/7. You deploy your code to them, they handle the servers, scaling, and uptime.

Example: Vercel, Netlify, and Railway are hosting platforms. You push code, they make it accessible at a URL.

HTML (Hypertext Markup Language) The basic structure language of websites. With modern frameworks like Next.js, you rarely write raw HTML - you write React components instead.

Example: `<button>Click me</button>` is HTML that creates a button.

I

Index / Indexing A database optimization that makes searches faster. Like the index in the back of a book - instead of reading every page to find something, you can look it up directly.

Example: If you search users by email frequently, the database creates an index on the email column to speed up those searches.

Integration Connecting your app to external services like Stripe for payments, SendGrid for emails, or Google for authentication.

Example: "Integrating Stripe" means adding code that communicates with Stripe's system to process payments.

J

JavaScript The programming language of web browsers. Most modern web apps are built with JavaScript or frameworks that use it. AI writes JavaScript for you - you just describe what you want.

Example: When you click a button and see something change on the page without reloading, that's JavaScript in action.

JWT (JSON Web Token) A secure way to keep users logged in by giving them a token their browser sends with each request. You never implement this manually - AI does it when you set up authentication.

Example: After you log in, your browser gets a JWT token it shows the server to prove "I'm still this user" on each page.

L

Landing Page A standalone web page designed for a specific purpose - usually marketing, signups, or product launches. Often the first page people see.

Example: A simple page with "My New App" headline, features list, and "Sign Up" button.

Load Balancing Distributing incoming traffic across multiple servers so no single server gets overwhelmed. Hosting platforms like Vercel handle this automatically.

Example: On Black Friday when everyone visits your store at once, load balancing splits visitors across multiple servers.

M

Markdown A simple way to write formatted text using plain text characters. Used for documentation, blog posts, and comments.

Example: `**bold**` *becomes* ***bold****,* `*italic*` *becomes italic. Much simpler than HTML.*

MDX Markdown with the ability to include interactive components. Used for rich blog posts and documentation that have more than just text.

Example: A blog post written in Markdown that includes a live calculator component.

Migration (Database) A script that changes your database structure - adding new columns, creating tables, or modifying relationships. Think of it as renovation instructions for your database.

Example: You realize users need a "bio" field, so you create a migration that adds a bio column to the users table.

MVP (Minimum Viable Product) The simplest version of your product that solves the core problem and can be tested with real users. Not feature-complete, just barely functional enough to be valuable.

Example: Instagram's MVP was just photo sharing with filters - no stories, reels, or messaging.

N

Netlify A hosting platform great for static websites and simple web apps. Free tier is generous. Similar to Vercel but originally focused on static sites.

Example: You push your portfolio site to GitHub, Netlify automatically deploys it and gives you a URL.

Next.js A modern React framework that makes building full-stack web apps easier. It handles routing, optimization, server-side rendering, and API routes. Recommended for Path 2 in this guide.

Example: Instead of setting up routing manually, Next.js automatically creates routes based on your file structure.

Node.js JavaScript that runs on servers (back-end) instead of just browsers (front-end). Lets you use JavaScript for your entire app.

Example: Your Next.js back-end runs on Node.js, even though you're just writing JavaScript.

O

OAuth A secure protocol for logging in with existing accounts like Google, GitHub, or Facebook. You never implement OAuth from scratch - services like Supabase handle it.

Example: "Sign in with GitHub" - you're redirected to GitHub, approve access, and return to the app logged in.

ORM (Object-Relational Mapping) A tool that lets you work with databases using code instead of writing SQL. Makes database operations feel more natural.

Example: Instead of writing SQL like `SELECT * FROM users WHERE email = ?`*, you write* `users.find({ email: 'test@example.com' })`*.*

P

Pagination Loading data in chunks instead of all at once. Like showing 20 results per page instead of 10,000 results that crash the browser.

Example: Google search shows 10 results per page with "Next" button - that's pagination.

PCI Compliance Security standards for handling credit card information. You never worry about this because Stripe handles all credit card data for you.

Example: By using Stripe Checkout, you're automatically PCI compliant because cards never touch your servers.

PostgreSQL A powerful, industry-standard database. This is what Supabase uses. Pronounced "post-gres-Q-L" or just "postgres."

Example: Your user data, posts, and comments all live in a PostgreSQL database.

Preview Deployment A temporary version of your app that gets created for each code change before it goes live. Let you test features before launching.

Example: You create a new feature on a branch, and Vercel gives you a preview URL to test it before merging to production.

Production The live, real version of your app that actual users are using. As opposed to "development" which is you testing on your computer.

Example: "The bug is in production" means real users are experiencing it right now on the live site.

Prompt The instructions you give to AI. In vibe coding, prompts describe what you want your app to do.

Example: "Create a sign-up form with email, password, and password confirmation" is a prompt for AI.

R

Railway A hosting platform that makes deploying Python apps and databases super easy. Good for Flask apps and PostgreSQL databases.

Example: You connect your GitHub repo to Railway, and it automatically deploys your Flask app and sets up a database.

React A JavaScript library for building user interfaces. Next.js is built on React. You describe what the UI should look like, React handles updating it.

Example: When data changes, React automatically updates the parts of the page that need to change without reloading.

Real-Time Features that update instantly without refreshing the page. Like seeing new messages appear in a chat as soon as someone sends them.

Example: Google Docs where you see other people's changes appear live as they type.

Render A hosting platform similar to Railway and Vercel. Good free tier for small projects.

Example: You can deploy a Flask back-end and PostgreSQL database on Render's free tier.

Resend An email service designed for developers. Makes sending transactional emails (welcome, password reset, notifications) very easy.

Example: When a user signs up, Resend sends them a welcome email from your app.

Responsive Design that adapts to different screen sizes - desktop, tablet, phone. A responsive site looks good on all devices.

Example: The navigation menu shows as a full bar on desktop but becomes a hamburger icon on phones.

REST API A standard way for front-end and back-end to communicate using HTTP requests. The most common API style.

Example: Your front-end sends a GET request to `/api/posts` and receives a list of posts as JSON.

Rollback Reverting to a previous version of your app when something breaks. Like undo for deployments.

Example: You deploy a new feature but it breaks login, so you rollback to the previous version in 30 seconds.

Route / Routing The system that determines what page shows up based on the URL. `/login` shows the login page, `/dashboard` shows the dashboard.

Example: In Next.js, creating a file at `app/about/page.js` automatically creates the `/about` route.

Row A single item in a database table. One row = one user, or one post, or one comment.

Example: In the users table, each row represents one person's account information.

S

SaaS (Software as a Service) Software you access through a web browser by paying a subscription. Like Netflix for business tools.

Example: Gmail, Notion, and Slack are SaaS products - you pay monthly and access them online.

Scaling Your app handling more users and traffic without slowing down or crashing. Good hosting platforms scale automatically.

Example: Your app works great with 100 users. Scaling means it still works great with 10,000 users.

Server A computer that runs your back-end code and responds to user requests. Hosting platforms manage servers for you.

Example: When you visit a website, your browser talks to a server that sends back the webpage.

Session The period of time a user is logged in. Sessions remember who you are as you navigate through an app.

Example: You log in once, and your session keeps you logged in as you click around for the next hour.

shadcn/ui A collection of pre-built UI components you can copy and paste into your app. Beautiful, accessible, and customizable.

Example: Instead of building a dropdown menu from scratch, you copy shadcn's dropdown component and adjust the styling.

Soft Delete Marking data as deleted without actually removing it from the database. Lets users undo deletions.

Example: Gmail's trash - emails are "deleted" but still exist for 30 days in case you want them back.

SQL (Structured Query Language) The language for talking to databases. With AI and modern tools like Supabase, you rarely write raw SQL.

Example: `SELECT * FROM users WHERE email = 'test@test.com'` *is SQL that finds a user by email.*

SSL Certificate What makes your website use "https" instead of "http" - it encrypts data between users and your site. Hosting platforms provide this automatically for free.

Example: The lock icon in your browser's address bar means SSL is working.

Stack The combination of technologies you use to build your app. "Next.js + Supabase + Vercel" is a stack.

Example: "What's your stack?" means "What tools/frameworks are you building with?"

Static Site A website where the content is the same for everyone. No personalization, no user accounts, no database. Fast and simple.

Example: A portfolio website or blog where every visitor sees the same pages.

Stripe The most popular payment processing service. Handles credit cards, subscriptions, invoices, and all payment complexity.

Example: When you buy something online and enter your card info in a Stripe checkout page.

Subscription A recurring payment model - users pay monthly or annually to keep access. Handled by Stripe.

Example: Netflix charges $15/month - that's a subscription.

Supabase An open-source Firebase alternative that gives you a PostgreSQL database, authentication, file storage, and real-time features in one service. Generous free tier.

Example: Instead of setting up separate services for database, auth, and storage, Supabase handles all three.

T

Table A collection of similar data in a database. Like a spreadsheet tab. You have a users table, posts table, comments table, etc.

Example: The "users" table stores all user accounts, with one row per user.

Tailwind CSS A CSS framework that works with utility classes. Instead of writing custom CSS, you apply classes like `bg-blue-500` and `text-center`.

Example: `<button className="bg-blue-500 text-white px-4 py-2 rounded">Click me</button>` *creates a styled blue button.*

Token A random string used to identify or verify something - like a login session or password reset request.

Example: When you reset your password, the link contains a token like `?token=a8b9c7d6e5f4` that identifies your reset request.

Transactional Email Automated emails triggered by user actions - welcome emails, password resets, purchase confirmations. Different from marketing emails.

Example: When you sign up for a service and immediately get a "Welcome!" email.

TypeScript JavaScript with type safety - the code knows what kind of data each variable should hold. Helps catch bugs but adds complexity. Optional for beginners.

Example: In TypeScript you declare that `age` must be a number, so it yells at you if you try to store text in it.

U

UI (User Interface) Everything users see and interact with. Buttons, forms, menus, text - the visual design of your app.

Example: "The UI needs work" means the app looks ugly or is hard to use.

URL (Uniform Resource Locator) A web address like `https://myapp.com/dashboard`. Each page in your app has a URL.

Example: `https://buildtolaunch.com/about` is the URL for the about page.

UUID (Universally Unique Identifier) A long random ID used to identify things in databases. Looks like `550e8400-e29b-41d4-a716-446655440000`.

Example: Instead of user IDs like 1, 2, 3 which are guessable, UUIDs are impossible to guess and globally unique.

UX (User Experience) How it feels to use your app. Good UX means users can accomplish their goals easily without confusion.

Example: "Great UX" means the app is intuitive and pleasant to use. "Bad UX" means users get lost or frustrated.

V

Validation Checking that user input is correct before accepting it. Like verifying an email address has an @ symbol.

Example: "Please enter a valid email" appears when you type "notanemail" into an email field - that's validation.

Variable A named container for storing data in code. Like a labeled box. You rarely deal with these directly - AI writes the code with variables.

Example: `let userName = "Jenny"` stores the name "Jenny" in a variable called userName.

Vercel The best hosting platform for Next.js apps (made by the same team). Free tier is generous, deployment is one command, auto-scales.

Example: You push code to GitHub, Vercel automatically builds and deploys it to a URL within minutes.

Version Control A system (usually Git) that tracks every change to your code so you can see history, undo changes, and collaborate with others.

Example: Like "Track Changes" in Word but for code - you can see exactly what changed and when.

Vibe Coding The method of building software by describing what you want to AI, which writes the code for you. You focus on vision and user needs, AI handles implementation.

Example: You say "users should be able to save bookmarks," AI generates all the code to make that happen.

W

Webhook A way for external services to notify your app when something happens. Like a doorbell that rings when an event occurs.

Example: When a payment succeeds on Stripe, Stripe sends a webhook to your app saying "payment completed for user X."

Workflow A sequence of steps or processes. Can refer to your development process or automated steps in your app.

Example: "My workflow is: describe feature to AI, test it, deploy" or "The checkout workflow is: cart → payment → confirmation."

X

XSS (Cross-Site Scripting) A security vulnerability where malicious code gets injected into your app. AI implements protections automatically when you use modern frameworks.

Example: Without XSS protection, someone could post a comment with malicious code that runs on other users' browsers.

Y

YAML A human-readable format for configuration files. Looks like organized lists with indentation. Used for some config files.

Example:

```yaml
name: My App
version: 1.0
features:
  - auth
  - payments
```

Terms You Can Safely Ignore

Some technical terms exist but don't matter for vibe coding. If you see these in documentation, don't worry about understanding them:

- **Binary trees** - Computer science data structure

- **Big O notation** - Algorithm efficiency measurement
- **Kubernetes** - Container orchestration (way overkill)
- **Microservices** - Complex architecture pattern
- **Webpack** - Build tool (frameworks handle this)
- **Babel** - JavaScript compiler (frameworks handle this)
- **Redux** - State management (use simpler options)

AI handles all of these automatically. Your job is describing what you want, not understanding how computers optimize things.

How to Use This Glossary

When building: If AI mentions a term you don't understand, look it up here. If it's not here, it's probably too technical and you can ask AI "explain that in simple terms."

When learning: Don't try to memorize these. Reference them as needed. You'll naturally learn the important ones through building.

When stuck: If an error message includes a term you don't know, check here first. Understanding the term often makes the error message much clearer.

Remember: You don't need to know all these terms to start building. You'll learn them as you go, when you actually need them.
