

The Complete Guide to Shipping with BUILD



Build

Contents

1. Introduction	3
2. Fundamentals	5
3. Getting Started	8
4. Features and Workflows	12
4.1 Conversational Mode	13
4.2 Design Mode	15
4.3 Figma Integration	17
4.4 Live Preview and Iterative Building	19
4.5 Deployment	21
4.6 Custom Domains and Netlify	23
4.7 Community Showcase	25
4.8 From BUILD to IDE	26
5. Use Cases	31
6. Tips, Patterns and Troubleshooting	34
7. Resources and Reference	36

CHAPTER 1

Introduction



1. Introduction

BUILD is CodeMate's browser-based AI studio for generating, previewing, and deploying full-stack applications. Accessible at build.codemate.ai, it generates a complete, runnable application including frontend, backend, database schema, authentication, and deployment configuration from a plain language description. The output is production-grade code that the team owns completely, with no licensing restrictions.

BUILD generates actual source code structured as a real project with proper folder organisation, typed components, and a working backend. The code can be downloaded as a ZIP, forked, or published to a live public URL, all from within the browser. Because the output is real code, anything started in BUILD can be continued in the IDE with C0 and CORA.

Who this guide is for

- Teams who want to go from idea to deployed application without local setup
- Design teams converting Figma files into working applications
- Developers who want to prototype fast, then continue in the IDE
- Anyone building internal tools, dashboards, or shareable demos

What you will learn

- What BUILD is and the four-stage workflow behind every project
- How to choose between Conversational, Design, and Figma Import modes
- How to use Live Preview, Go Live, and custom domains
- How to move a BUILD project into the IDE and continue with C0 and CORA
- Common issues and how to resolve them

CHAPTER 2

Fundamentals



2. Fundamentals

What BUILD is

BUILD takes any team from a description to a deployed, running full-stack application without local setup, framework configuration, or manual scaffolding. It generates real production-grade code that can be published instantly or taken into the IDE to continue building with C0 and CORA. BUILD is suited for product teams validating new ideas, design teams turning approved mockups into code, and engineers who need a complex scaffold without days of setup.

The BUILD workflow

Every BUILD project follows four stages:

- **Describe.** Define the application in plain language. Attach reference files, screenshots, or a PRD using the + button to give BUILD additional context.
- **Generate.** BUILD plans the architecture and creates the complete codebase, with files appearing in real time.
- **Preview and Iterate.** The application runs in a live preview. Follow-up prompts make changes immediately.
- **Go Live.** Click Preview site, then Go Live to publish at a public URL with HTTPS configured automatically, or take the code into your own environment by downloading or forking it.

Core concepts

- **Real code, not templates.** BUILD generates actual source code with proper folder organisation and typed components not pre-built templates or low-code blocks.
- **Full ownership.** The output is production-grade code the team owns completely, with no licensing restrictions.
- **Three starting modes.** Conversational, Design, and Figma Import. The mode is chosen at the start of a project and cannot be switched mid-project.
- **Seamless handoff to the IDE.** Because the output is real code, any BUILD project can be forked, downloaded, and continued with C0 and CORA in VS Code, JetBrains, Cursor, or Antigravity.

CHAPTER 3

Getting Started



3. Getting Started

Accessing BUILD

Navigate to **build.codemate.ai** in any modern browser, or click Build in the CodeMate Web Application sidebar, or click Open on the BUILD card in Toolbox. No installation is required. Sign in with your CodeMate account and you land directly on the BUILD home screen.

Choosing a starting mode

Modes cannot be switched mid-project

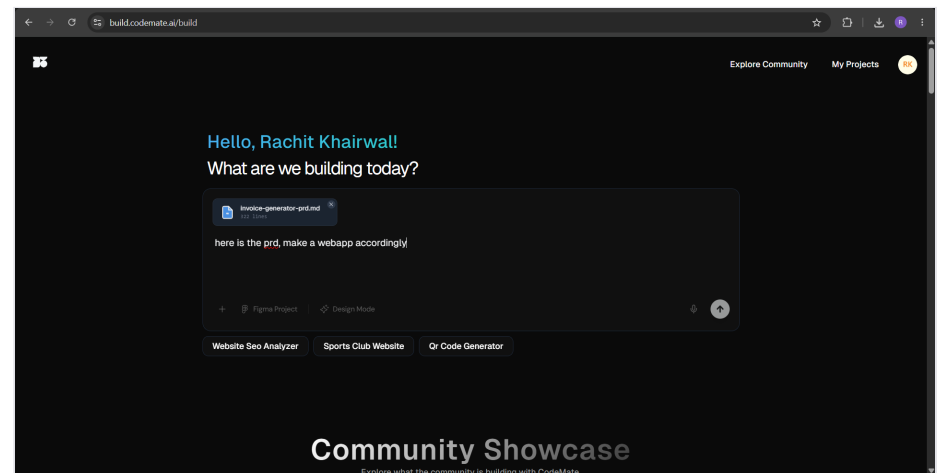
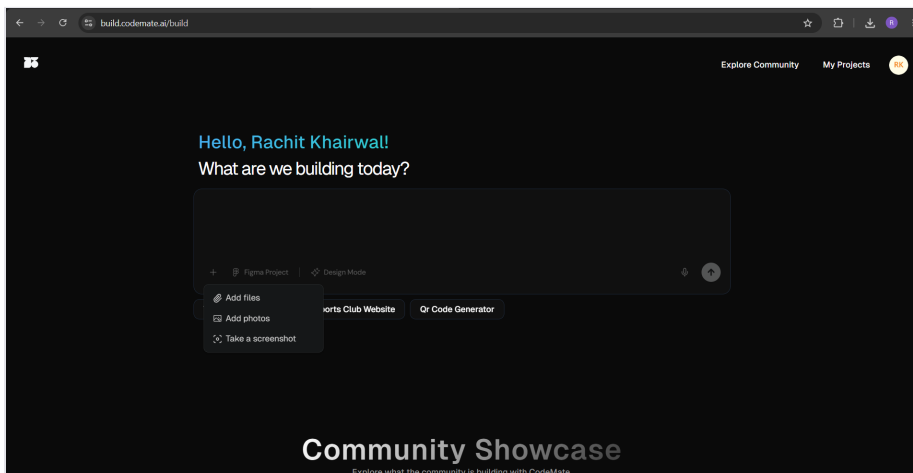
You select a starting mode when creating a project. To use a different mode, start a new project.

- **Conversational Mode.** The default when no toggle is active. Describe the application in plain language and BUILD generates the complete codebase.
- **Design Mode.** Activated by clicking the Design Mode toggle. BUILD generates visual design variations from your description for review before any code is written.
- **Figma Import Mode.** Activated by clicking the Figma Project toggle. Paste a Figma file URL and BUILD converts the design directly into a working application.

Attaching files, photos, and screenshots

In any mode, click the + button in the chat input to attach supporting material alongside your prompt. Three options are available:

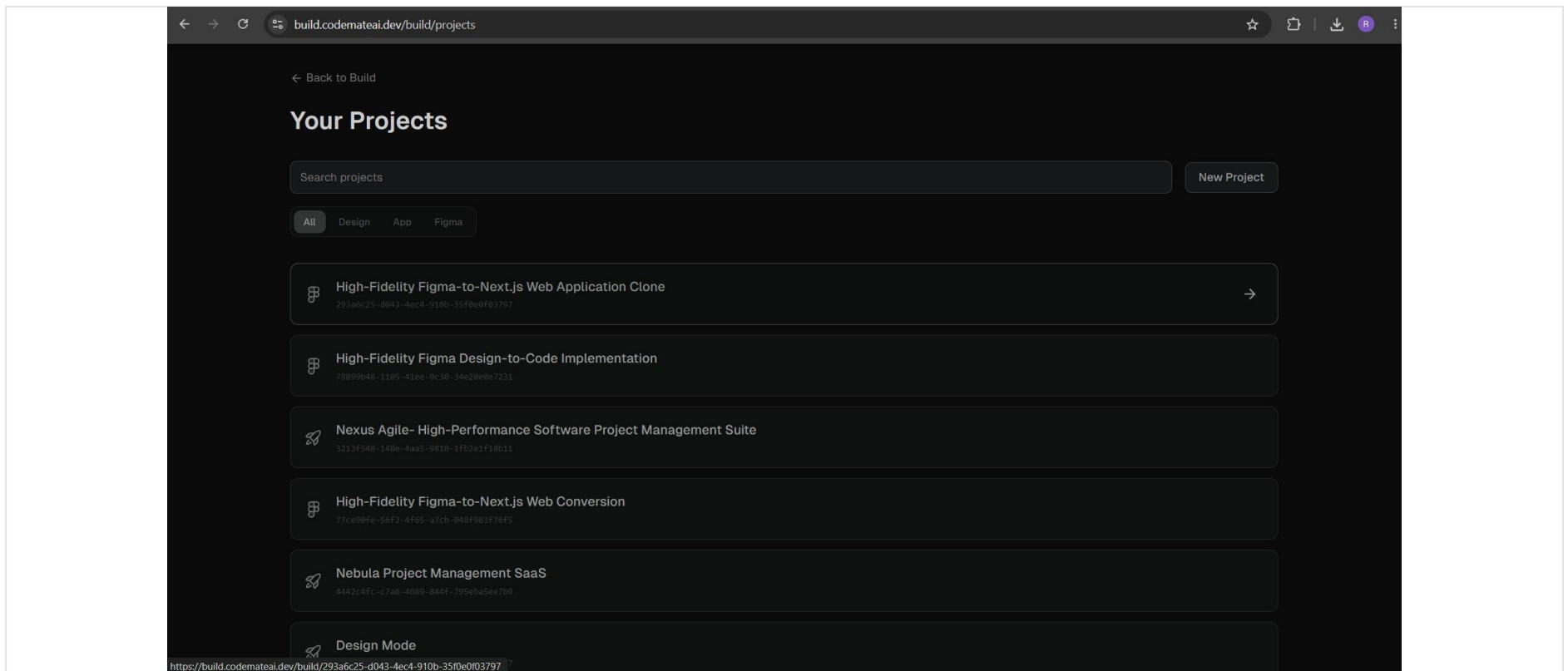
- **Add files** attach documents, Markdown files, code files, or any text-based reference such as a PRD or spec sheet.
- **Add photos** attach images or screenshots to use as visual reference or to clone an existing UI.
- **Take a screenshot** capture your screen directly without leaving BUILD.



The + menu (left) and a Markdown PRD attached with the prompt "here is the prd, make a webapp accordingly" (right)

Managing your projects

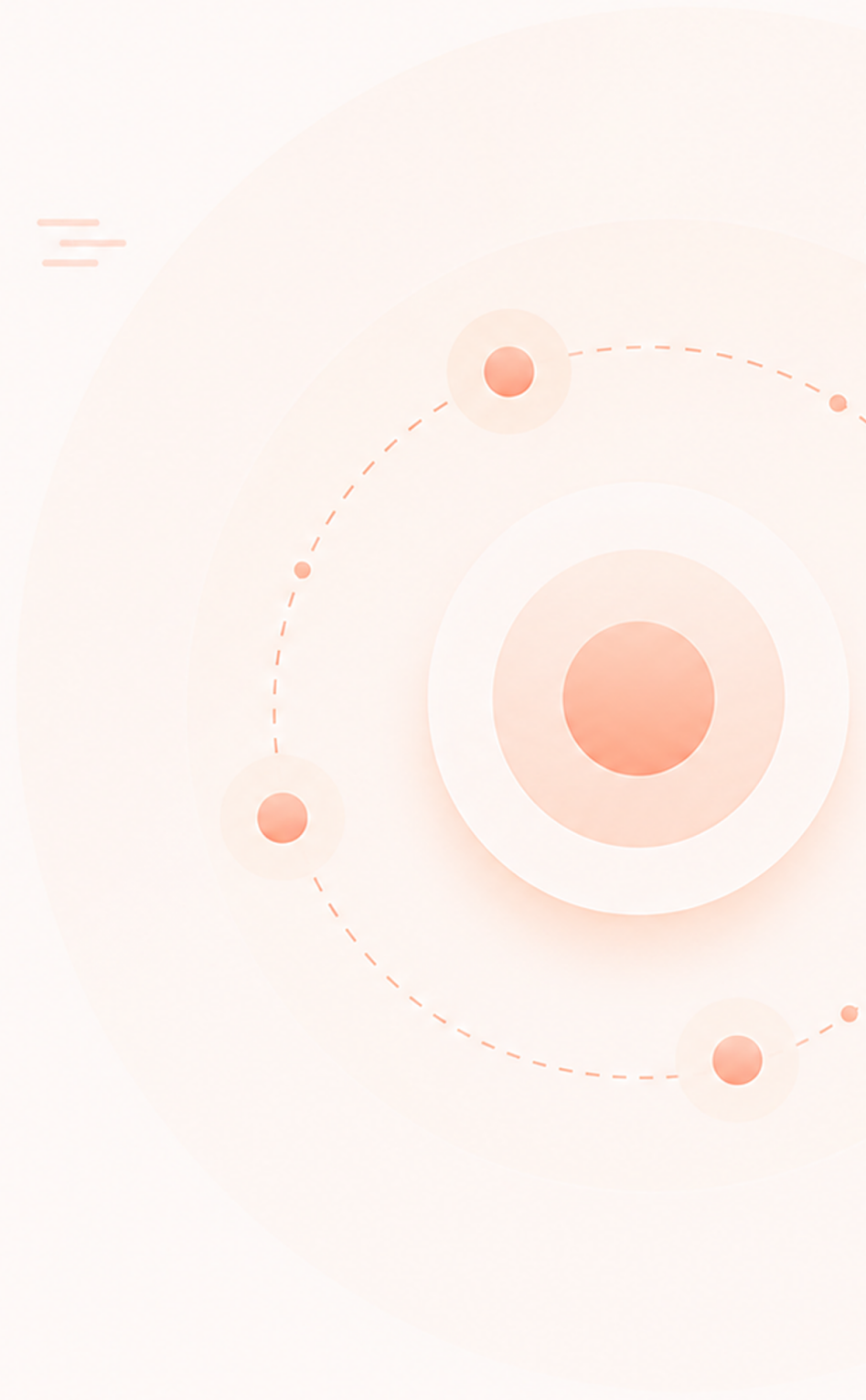
All previously created projects are accessible from the My Projects page via My Projects in the top navigation. Projects can be filtered by All, Design, App, and Figma type, and each opens in the BUILD editor with its full conversation history, file explorer, and preview.



The My Projects page showing previously created projects with search and filter functionality across all three types, with a New Project button in the top right

CHAPTER 4

Features & Workflows



4. Features and Workflows

4.1 Conversational Mode

Conversational Mode is the primary way to use BUILD. Describe the application in plain language and BUILD generates a complete, runnable codebase. Each follow-up prompt adds to or modifies the existing code incrementally rather than regenerating from scratch.

When to use it: Use when you know what you want to build and want to describe it in words. It is the fastest path from description to running application and works for any project from simple tools to complex multi-role applications.

How to use it: With no toggle active, type your application description and press Enter. Use the + button to attach files, photos, or screenshots alongside your prompt. BUILD generates an architecture plan and begins creating files in real time. Send follow-up prompts to make changes or add features.

Attaching files and images: You can attach a UI screenshot to have BUILD match or clone that design, attach a requirements document so BUILD builds exactly what is specified, or attach a reference image to define the visual direction. Files, photos, and screenshots are all supported via the + button.

Use Case Building from a PRD

A product manager attaches a Markdown PRD and prompts BUILD to generate the application from it. BUILD reads the document, plans the architecture, and generates a complete codebase matching the specified features and user flows.

Example prompt

Here is the PRD build a web app accordingly, with the user roles and flows described in the document.

Use Case Cloning an existing UI

A developer attaches a screenshot of an existing interface and asks BUILD to recreate it as a working application. BUILD reads the visual layout and generates matching components, colours, and structure.

Example prompt

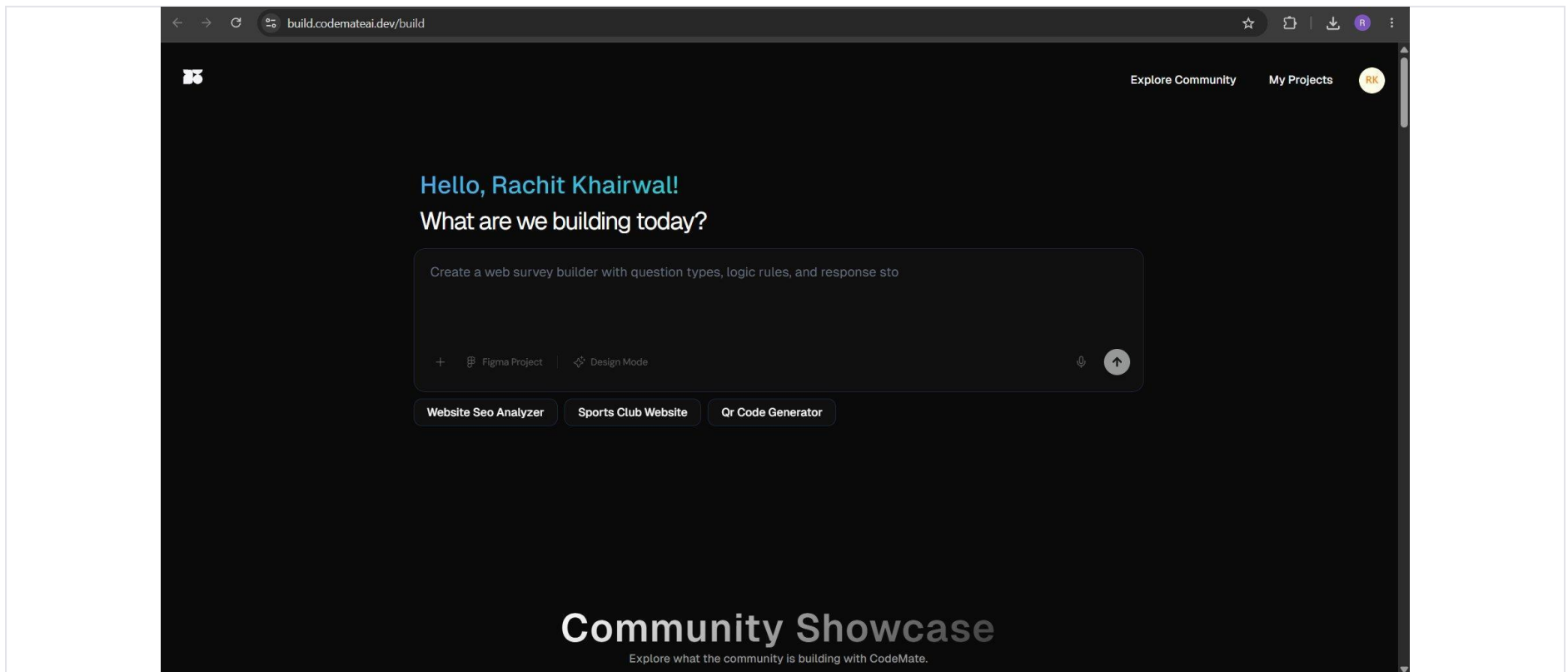
Here is a screenshot of the UI I want recreate this as a React application with a working backend and authentication.

Use Case Survey builder with analytics

A team describes a web survey builder with multiple question types, conditional logic, response analytics, and CSV export. BUILD generates a complete application with a survey builder interface, response collection, an analytics dashboard, and an authentication system separating creator and respondent roles.

Example prompt

Build a web survey builder with multiple question types, conditional logic, response analytics, and CSV export. Separate creator and respondent roles with authentication.



The BUILD home screen with a detailed conversational prompt, Figma Project and Design Mode toggles, and the + button for attaching files and photos

4.2 Design Mode

Design Mode generates high-fidelity visual designs from a text description before any code is written. BUILD produces rendered variations for review and refinement, then generates a full application matching the approved design.

When to use it: Use when visual quality is a primary concern, when multiple visual directions need evaluating, when the brief includes specific visual requirements such as layout, colour scheme, typography, or component style, or when you want to refine an existing design before generating code.

How to use it: Click the Design Mode toggle, describe the interface including layout, colour scheme, typography, and components, and press Enter. Review the variations, refine with follow-up prompts, then export to the code pipeline. You can also attach an existing design using the + button and prompt BUILD to enhance or iterate on it for example, adjusting the layout, updating font styles, or adding new components.

Use Case SaaS analytics dashboard

A product team describes a SaaS analytics dashboard with a dark theme, blue accents, four KPI cards, a revenue chart, and a notifications panel. BUILD generates two variations side by side. After a follow-up prompt requesting larger KPI cards and a date range picker, the approved design is exported to code and BUILD generates a fully functional application matching it exactly.

Example prompt

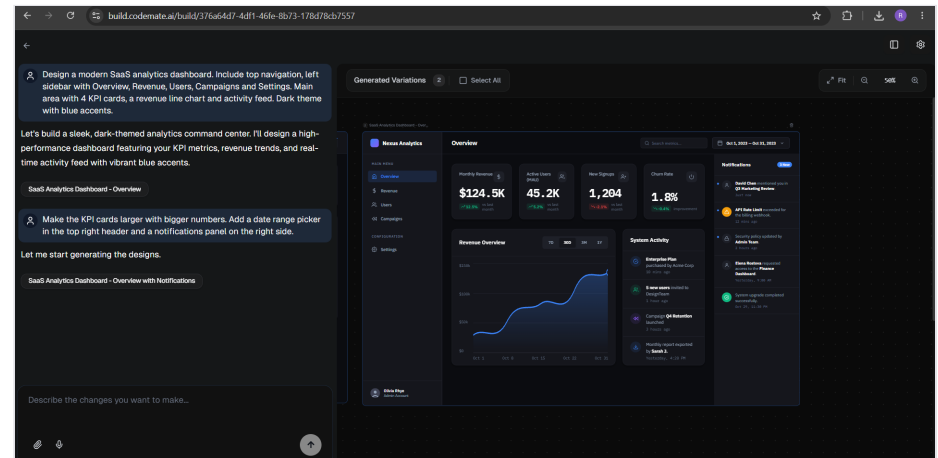
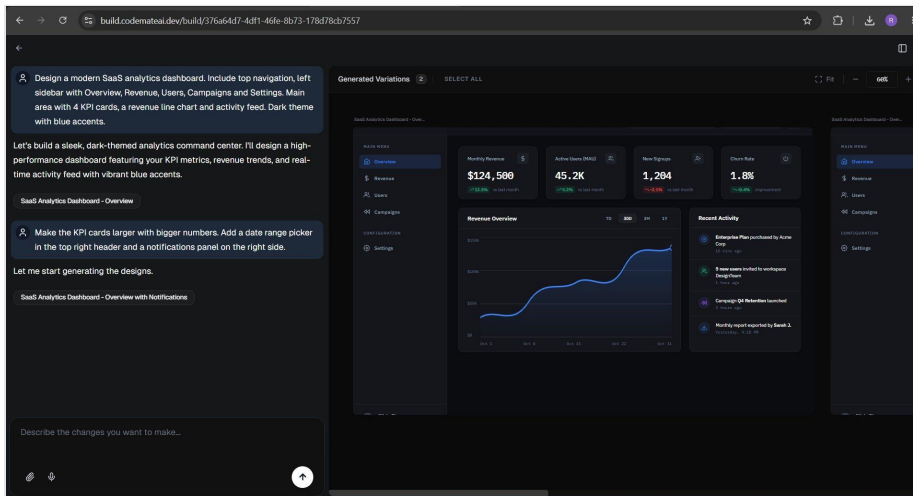
Design a SaaS analytics dashboard. Dark theme with blue accents. Top navigation, left sidebar with Overview, Revenue, Users, Campaigns and Settings. Four KPI cards, a revenue line chart, and a real-time activity feed.

Use Case Enhancing an existing UI design

A designer attaches a screenshot of a current interface and asks BUILD to improve it. Using the + button alongside Design Mode, they upload the design and prompt BUILD to refine the layout, update the typography, and modernise the colour palette. BUILD generates improved variations based on the input.

Example prompt

Here is our current dashboard UI modernise the layout, switch to a clean sans-serif font, and update the colour scheme to dark navy with green accents.



Design Mode generating two variations of a SaaS analytics dashboard the initial output (left) and a refined version with larger KPI cards, a date range picker, and a notifications panel (right)

4.3 Figma Integration

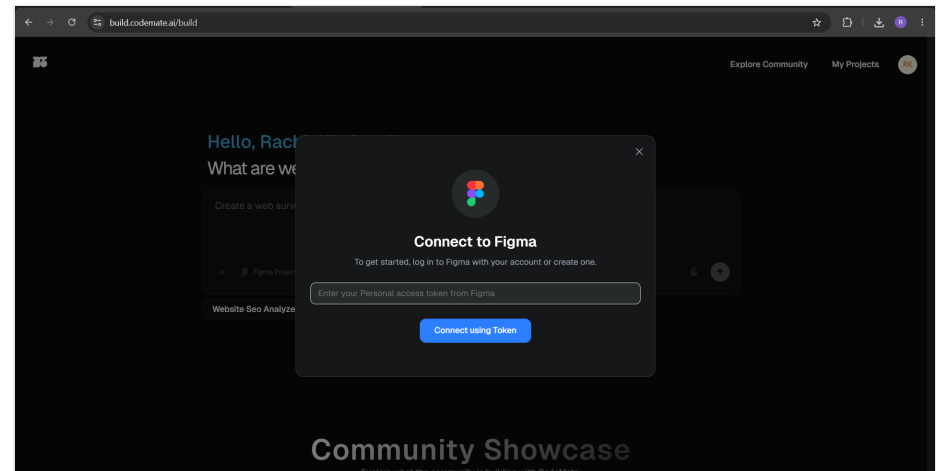
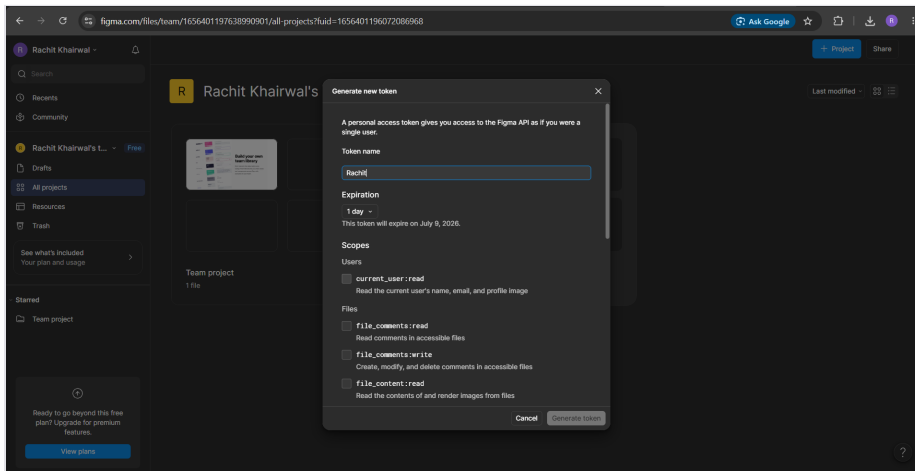
Figma Integration converts any Figma design file directly into a working full-stack application, reading all components, layouts, colours, typography, and assets, and generating a complete application with high fidelity using React components and Tailwind CSS.

When to use it: Use when a design already exists in Figma and needs to be converted to code, when approved mockups need implementing, or when high design fidelity is required from an existing source file.

Connecting Figma with a Personal Access Token

Before importing a Figma file, you need to connect your Figma account using a Personal Access Token (PAT). This gives BUILD read access to your Figma files via the Figma API.

Step 1 Generate the token in Figma: In Figma, click your profile icon, go to Settings, scroll to Personal access tokens, and click Generate new token. Give the token a name, set an expiration, and enable all scopes: `current_user:read`, `file_comments:read`, `file_comments:write`, and `file_content:read`. Click Generate token and copy it immediately it is shown only once.



Generating a PAT in Figma with all scopes enabled (left) and pasting it into the Connect to Figma modal in BUILD (right)

Step 2 Connect in BUILD: On the BUILD home screen, click the Figma Project toggle. A Connect to Figma modal appears with a text field labelled "Enter your Personal access token from Figma." Paste the token and click Connect using Token.

Step 3 Import the design: After connecting, the Choose a Figma frame dialog opens. Paste your Figma frame or file URL and click Start Import. BUILD reads the design and generates the complete application. For best results, use Auto Layout throughout your Figma file and name layers clearly.

Reconnecting: If your token expires, click Reconnect Figma in the import modal to re-authenticate with a new token.

Use Case Converting an approved product design

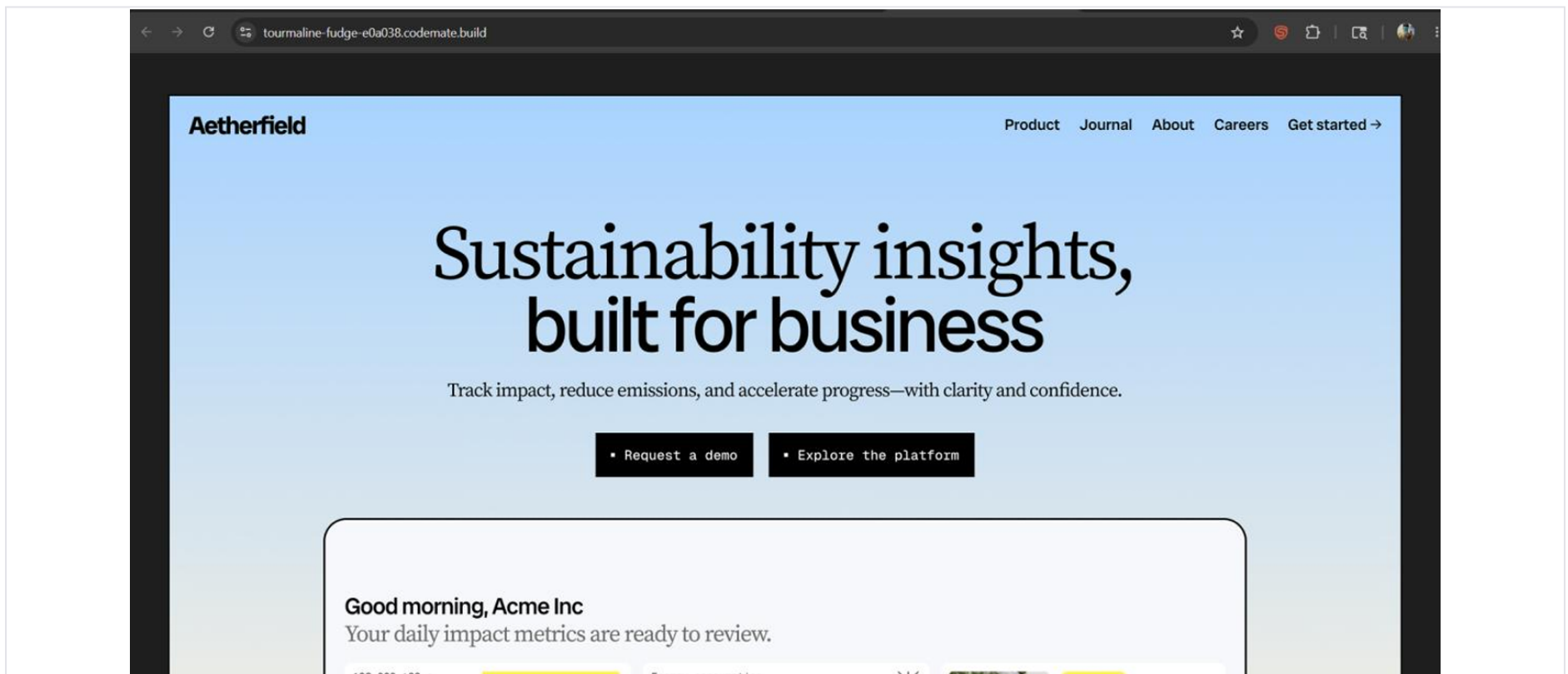
A design team has a fully approved Figma file for a new SaaS product. Rather than handing off specs to engineering, they paste the Figma URL into BUILD. BUILD reads every component, layout, colour, and typography style and generates a pixel-accurate React application using Tailwind CSS ready to review, iterate on, and deploy.

Example prompt (after import)

The design is imported now add form validation to the signup flow and connect the dashboard to the mock API endpoints defined in the spec.

Use Case iOS-style component library to web app

A team has a Figma file containing an iOS-style component library with glassmorphism effects. They paste the URL into BUILD, which generates a complete Next.js application using React components, Tailwind CSS, and shadcn/ui overrides that faithfully reproduce the blur, liquid glass, and transparency effects from the source design.



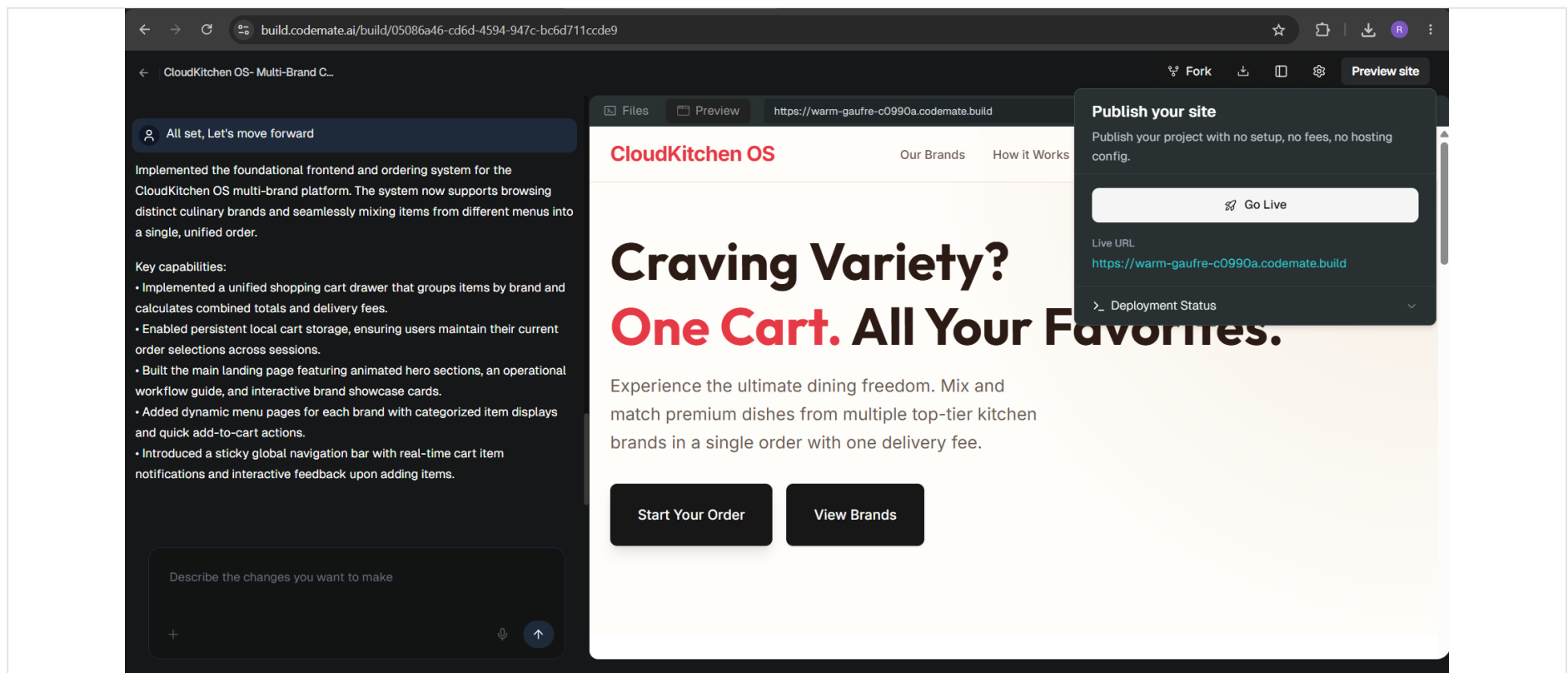
A live application generated from a Figma import the Aetherfield sustainability insights dashboard, deployed at a codemate.build URL with full navigation and dashboard components

4.4 Live Preview and Iterative Building

The live preview runs the generated application directly in BUILD, updating in real time as follow-up prompts are processed. It lets you test the application as it evolves without leaving the browser.

When to use it: Use continuously to verify the application works, test user flows, check responsive layouts, and catch issues before they are baked into the codebase.

How to use it: After generation, the preview loads automatically in the Preview tab. Click through the application to test it, then type any change in the chat input and press Enter. BUILD applies the change and the preview updates immediately. Use follow-up prompts to add pages, adjust layouts, change components, or fix issues you spot in the preview.



The BUILD live preview showing the application running in the browser alongside the Publish your site panel click Go Live to make the application accessible to anyone via a public URL

Use Case Iterating on a project management tool

A developer reviewing a generated project management tool notices the task list lacks due dates and the colour coding for overdue tasks is missing. They send a follow-up prompt to add due date display with overdue tasks highlighted in red. BUILD applies the change and the preview updates. A second prompt refines the visual treatment immediately.

Example prompt

Add due date display to each task card. Highlight overdue tasks in red. Show a summary count of overdue tasks at the top of the list.

Use Case Adding additional pages

After reviewing the initial generation, a team wants to add a settings page and an onboarding flow that were not in the original prompt. They describe each new page in a follow-up and BUILD adds it to the existing codebase, matching the established component patterns and routing structure.

Example prompt

Add a Settings page with sections for profile, notifications, and billing. Add a three-step onboarding flow that runs on first login and can be skipped.

4.5 Deployment

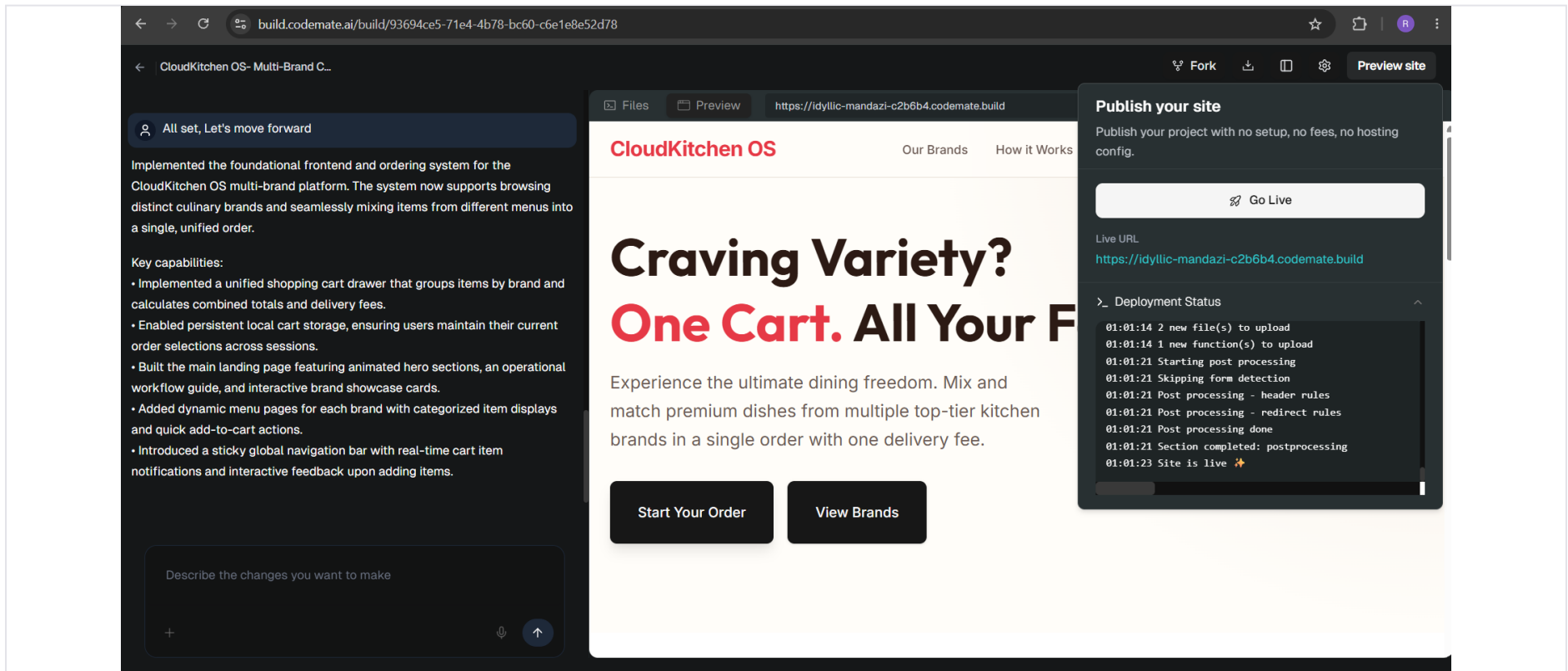
BUILD publishes generated applications directly to CodeMate Cloud, configuring HTTPS automatically and hosting at a public URL in the format `your-app-name.codemate.build`. Once live, anyone can open the application from any device.

When to use it: Go live to share the application with stakeholders, get a shareable link for an MVP or prototype, or prepare for a first round of testing outside the browser preview.

How to use it:

- Click **Preview site** in the top right of the BUILD editor. The **Publish your site** panel opens on the right.

- Click **Go Live**. BUILD publishes the application to CodeMate Cloud and the Deployment Status log updates in real time.
- Once the log shows **Site is live**, the Live URL is active and accessible from any browser or device.



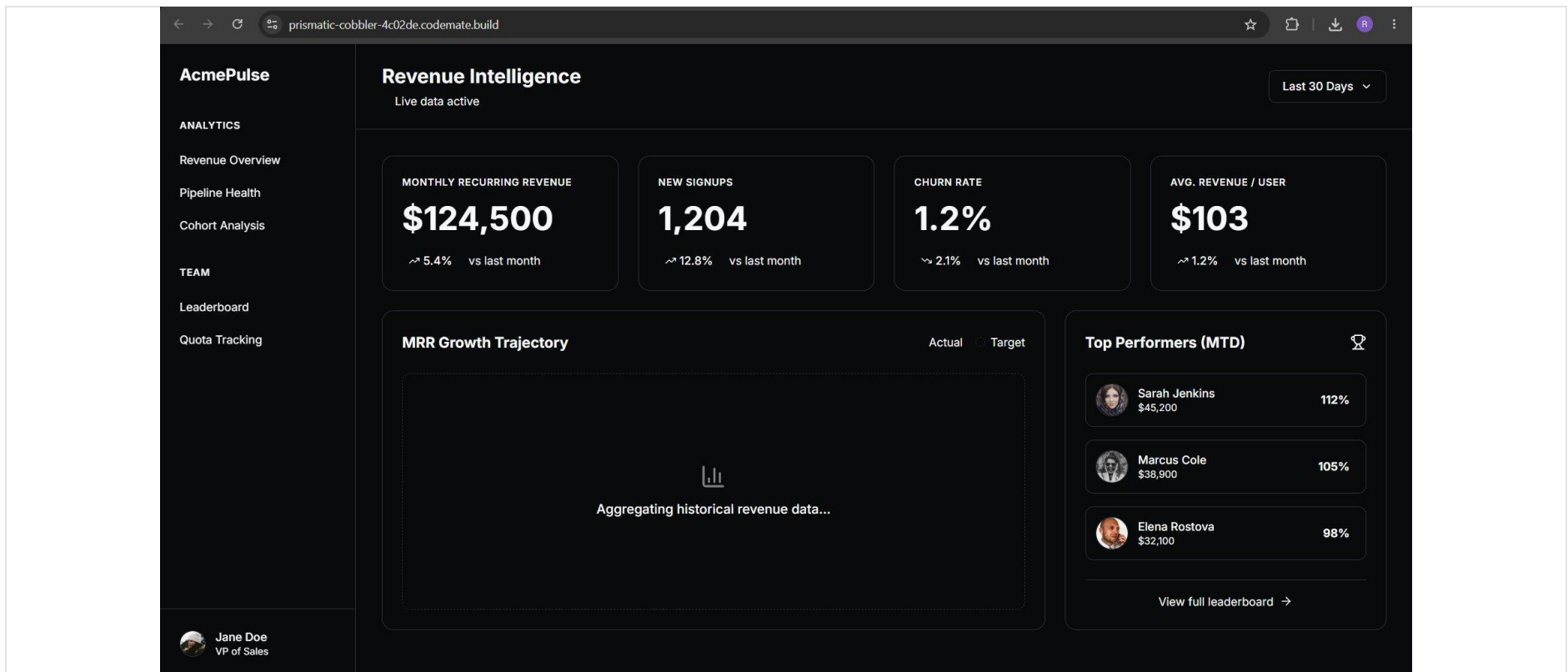
The Publish your site panel showing the Go Live button, the assigned Live URL, and the real-time Deployment Status log ending with "Site is live"

Use Case Sharing an MVP before a customer meeting

A solutions engineer builds a B2B analytics dashboard in BUILD, reviews it in the live preview, then clicks Preview site and Go Live. Once the Deployment Status shows Site is live, they copy the URL and share it ahead of a customer call. The customer opens a fully functional dashboard on their own device before the meeting begins.

Example prompt before going live

Add a loading skeleton to the KPI cards and ensure the revenue chart renders correctly on mobile before I share this with the client.



The deployed dashboard running live at a `codemate.build` URL a fully functional B2B SaaS analytics application with sidebar navigation, KPI cards, a growth chart, and a leaderboard

4.6 Custom Domains and Netlify

Every project gets a live URL the moment it goes live, assigned a subdomain under `.codemate.build` by default. For production-ready projects, BUILD gives two paths: a custom subdomain on `.codemate.build`, and a full transfer to Netlify for advanced deployment control and custom domain names.

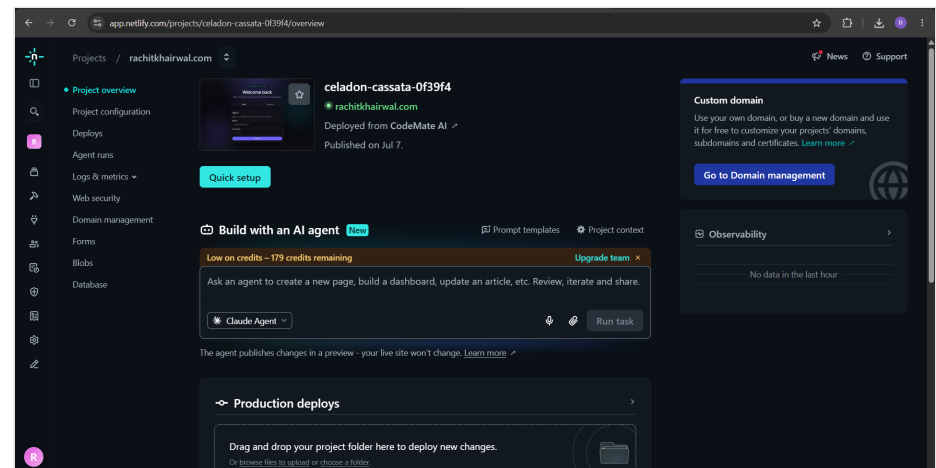
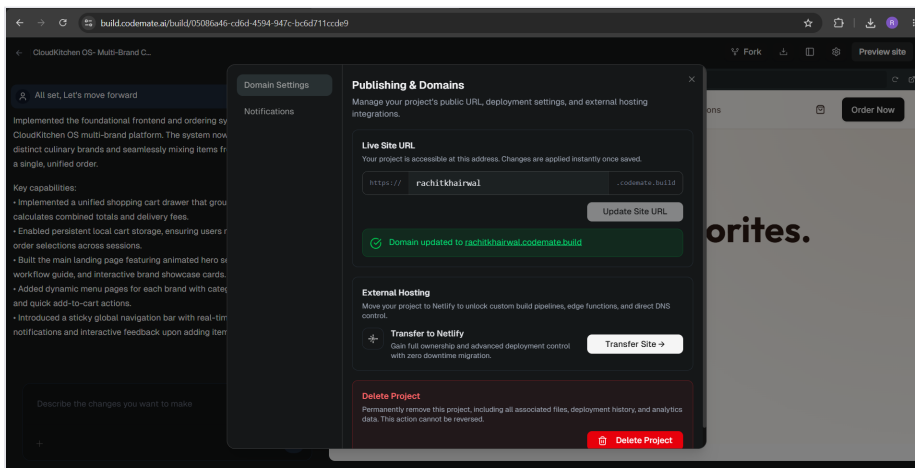
Setting a custom subdomain

- Click the **settings (gear) icon** in the top right of the BUILD editor to open the **Publishing & Domains** panel.
- Under **Live Site URL**, clear the existing subdomain and type your preferred one, keeping it lowercase and hyphen-separated.
- Click **Update Site URL**. A confirmation message appears and the project is immediately live at your chosen subdomain no redeployment needed.

Transferring to Netlify

For teams that need custom domain names, CI/CD pipelines, rollbacks, or advanced hosting control, BUILD supports a direct transfer to Netlify. This moves the project into your own Netlify account with zero downtime.

- In the Publishing & Domains panel, scroll to **External Hosting**.
- Click **Transfer Site** next to the Transfer to Netlify option.
- Authenticate with your Netlify account and confirm the transfer.
- Your project appears in your Netlify dashboard with CodeMate AI listed as the deployment source.



The Publishing & Domains panel for setting a custom subdomain and transferring to Netlify (left), and the Netlify project overview after transfer showing custom domain management and the AI agent builder (right)

What you get inside Netlify

After the transfer, the project is available at the same `.codemate.build` URL until a custom domain is configured. From the Netlify dashboard:

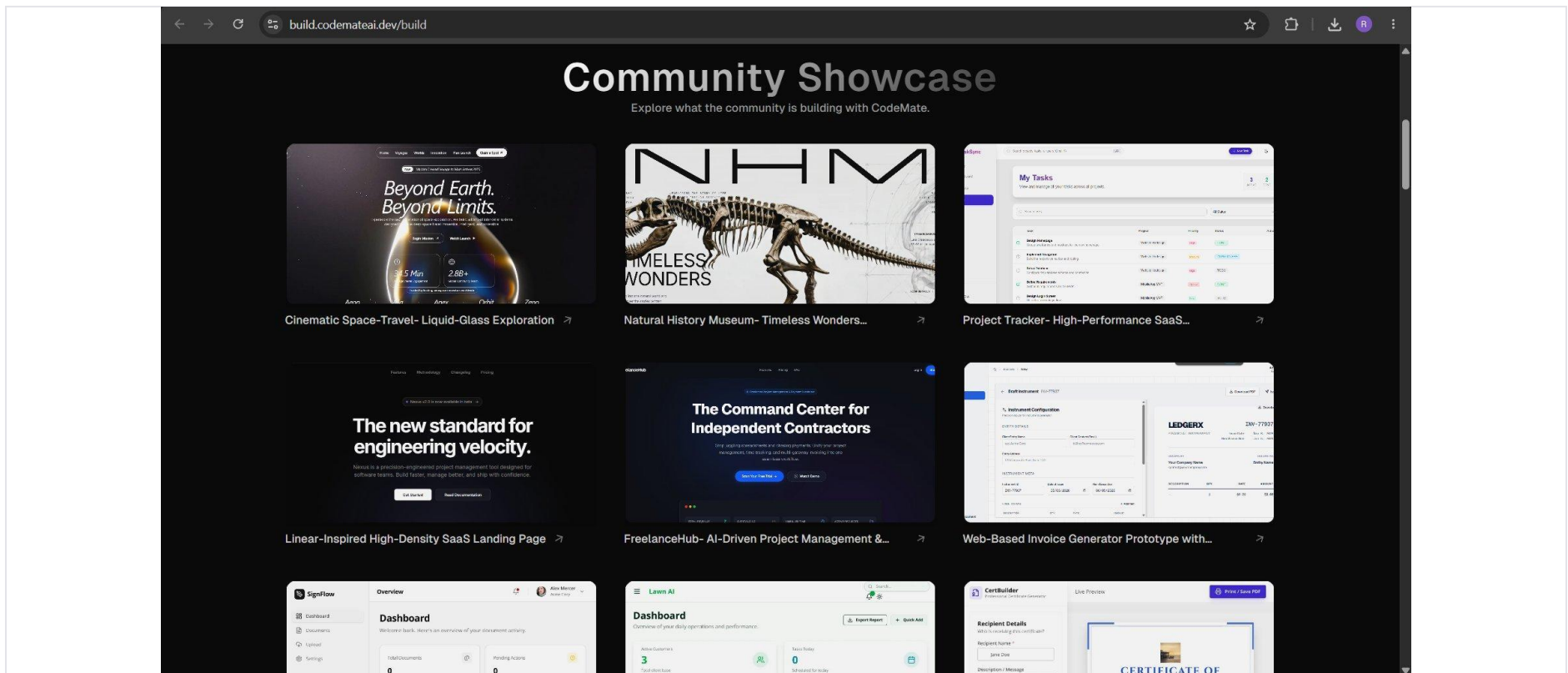
- **Project Overview** live URL, deployment source, last publish timestamp, and quick setup.
 - **Domain Management** connect a custom domain with automatic SSL and subdomain routing.
 - **Production Deploys** tracks every deployment; drag and drop a project folder to push new changes, or browse files to upload.
 - **Agent runs** a Claude-powered AI agent can make changes directly to the deployed project, published to a preview URL so the live site is never affected until approved.
 - **Logs & Metrics** monitors request counts, error rates, bandwidth, and function invocations.
 - **Project Configuration** build settings, environment variables, continuous deployment, post-processing, and build plugins.
-

4.7 Community Showcase

The Community Showcase is a public gallery of applications built and shared by the CodeMate community. Each entry shows a preview and a link to the live deployed version, and any showcase project can be forked into your account to use as a starting point.

When to use it: Use for inspiration, to evaluate what is possible with BUILD, or to find a project close to what you want to build and fork it rather than starting from scratch.

How to use it: Scroll to the bottom of the home screen or click Explore Community in the navigation. Click any project to open the live application, or fork it to create your own editable copy.



The BUILD Community Showcase displaying real applications built by the CodeMate community including a Cinematic Space Travel website, Natural History Museum, Project Tracker SaaS, FreelanceHub, Invoice Generator, SignFlow, Lawn AI, and CertBuilder

Use Case Forking a showcase project as a starting point

A developer building a freelance management platform finds a similar project in the showcase. Rather than starting from scratch, they fork it into their account and use follow-up prompts to adapt it changing branding, adding a contracts module, and removing features that do not apply.

The fork preserves the original showcase project intact.

Example prompt after forking

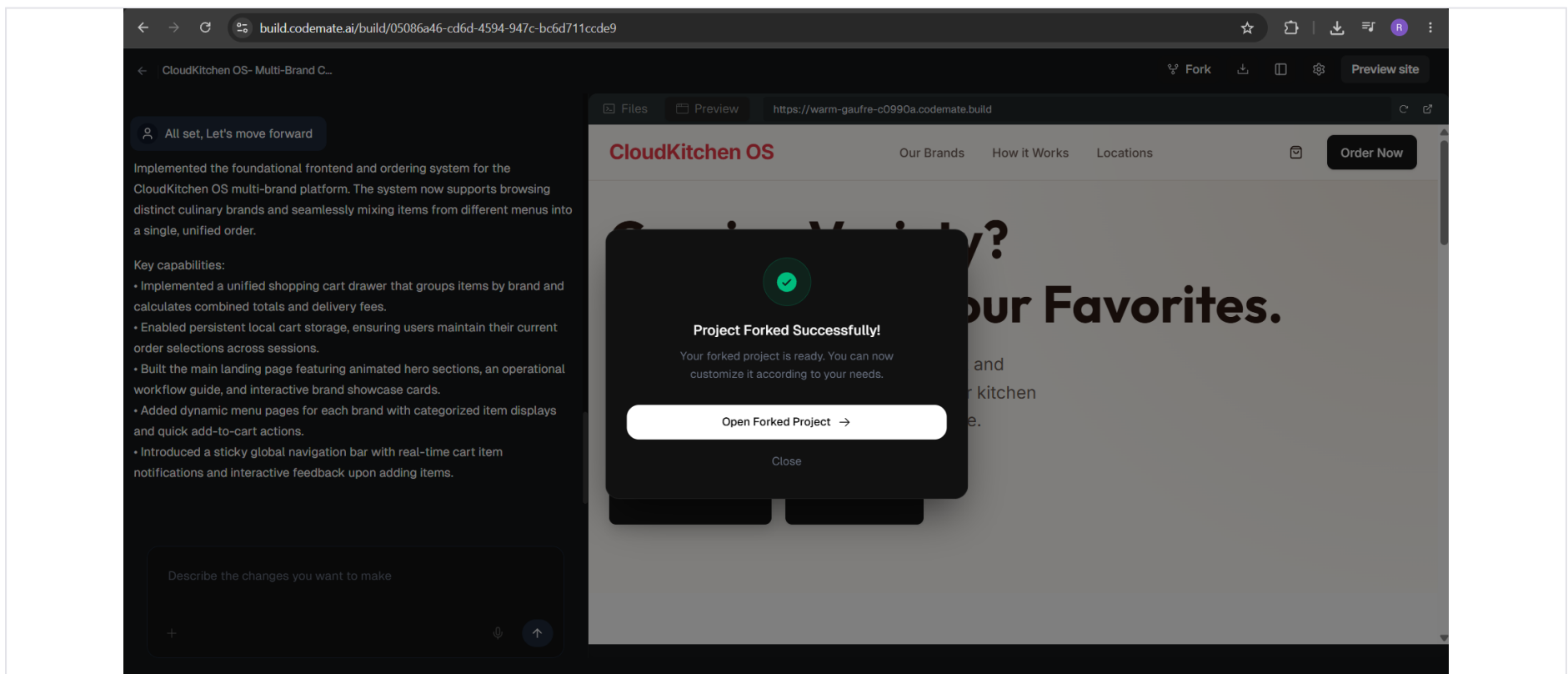
Remove the invoicing section, rebrand to "FreelanceOS", and add a contracts module where clients can sign proposals digitally.

4.8 From BUILD to IDE: Fork, Download, and Continue in Code

BUILD is the fastest way to get a working application, but most real projects eventually need work beyond what a browser studio is designed for: connecting to internal systems, adding complex business logic, or integrating with existing repositories. Because BUILD generates real code, this transition is seamless.

Step 1: Fork the project (optional)

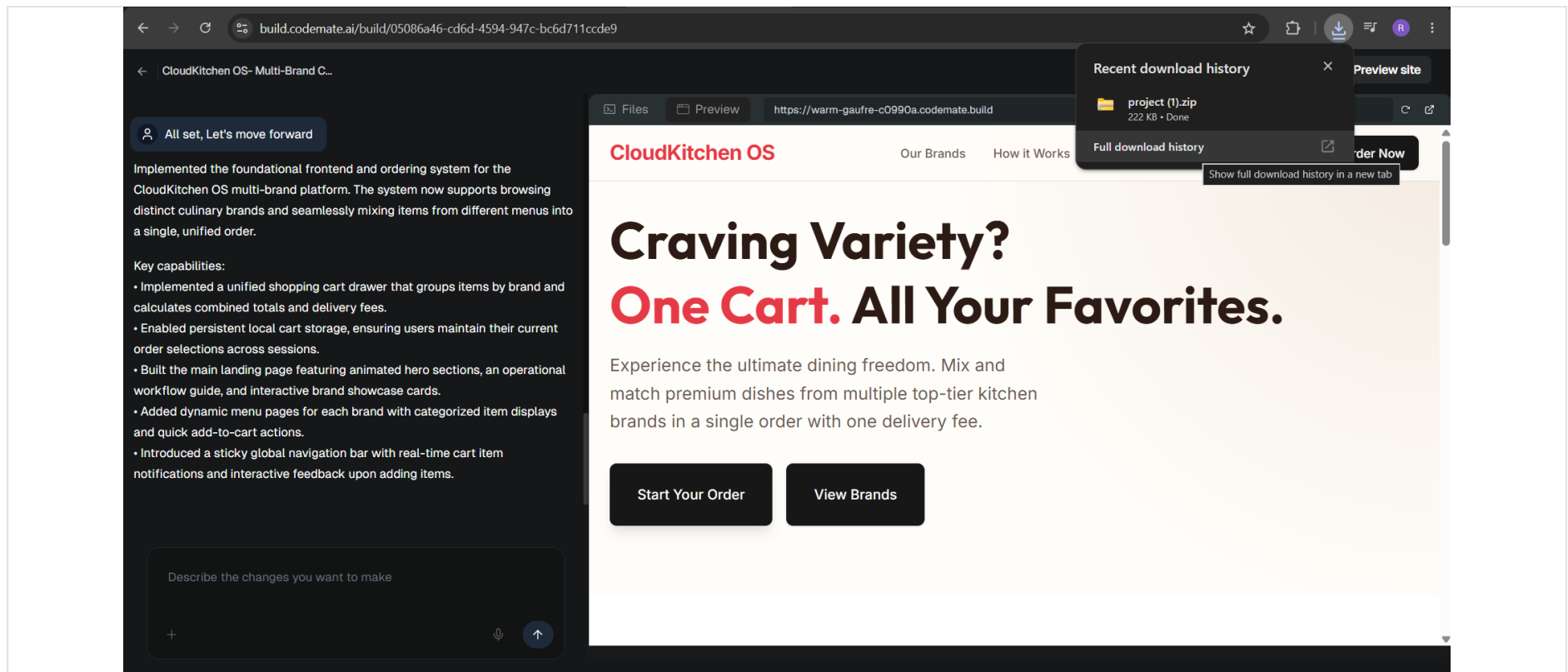
Forking creates an independent copy of a BUILD project so you can branch off in a new direction without affecting the original. This is useful when you want to keep the original deployed version intact while experimenting, or when starting from a community project. Click the **Fork** button in the top right of any project.



The BUILD Fork confirmation showing the Project Forked Successfully message with an Open Forked Project button, created from the CloudKitchen OS multi-brand platform project

Step 2: Download the code as a ZIP

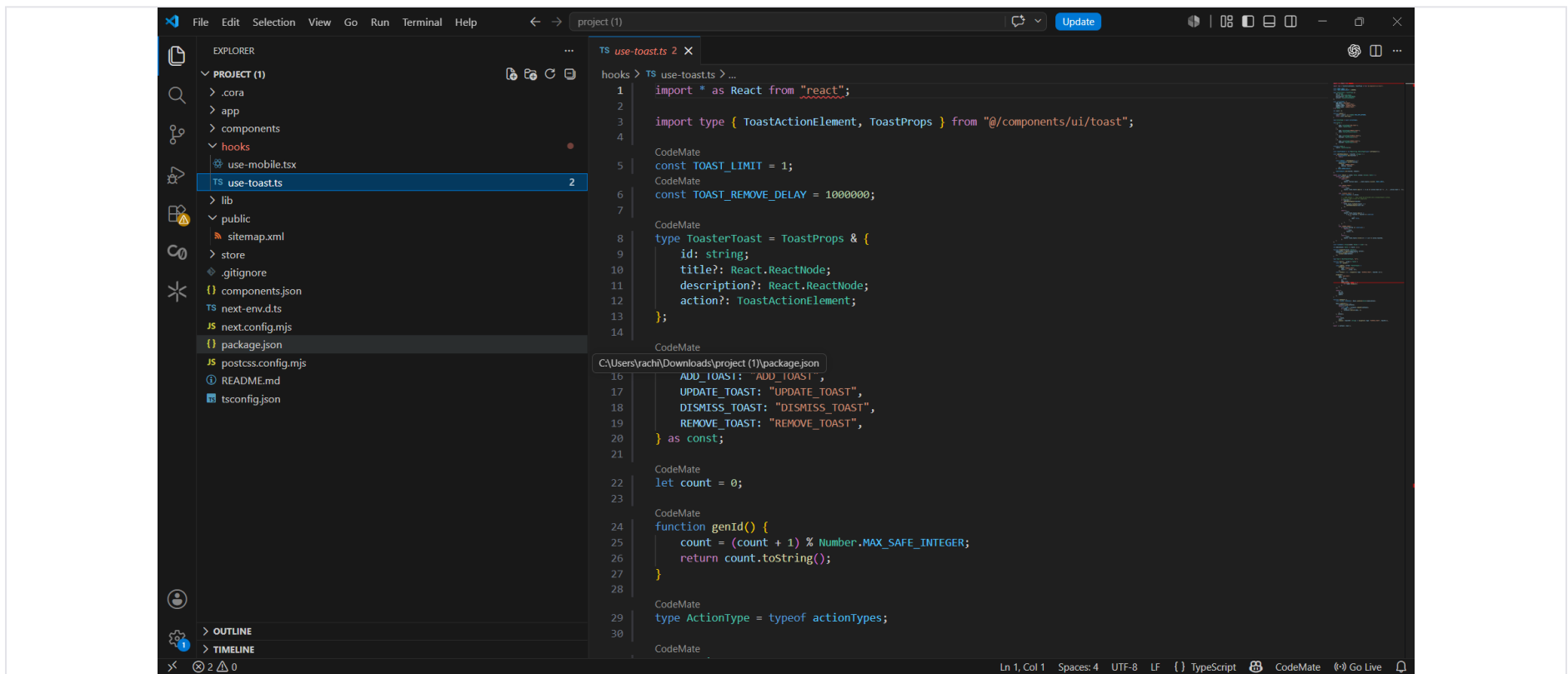
To move the project into your own development environment, click the **Download icon** in the top right of the project header. BUILD packages the complete project all source files, configuration files, and an environment variable template into a single ZIP archive ready to open in any IDE.



The BUILD download icon in the project header clicking it packages the complete project into a ZIP ready to open in your preferred IDE

Step 3: Open the project in your IDE

Unzip the downloaded archive and open the project folder in your preferred IDE. BUILD-generated projects work with **VS Code**, **JetBrains**, **Cursor**, and **Antigravity**. The full file structure application source, configuration files, and environment template is immediately visible in the file explorer.



The downloaded BUILD project opened in VS Code showing the complete generated file structure in the Explorer, including app, components, hooks, lib, public, and store folders

Step 4: Continue building with C0 and CORA

Once the project is open in your IDE with the C0 and CORA extensions installed, the full power of both assistants is available on the generated code. Because the Knowledge Base indexes the project automatically, C0 immediately understands the structure BUILD generated and can

answer questions about it, debug it, review it, optimise it, and add tests. CORA can plan and implement entire new features across the codebase in the correct order, matching the patterns BUILD established.

A productive approach is to use C0 to understand and improve what already exists, and CORA to add substantial new functionality that spans multiple files.

Use Case Extending a community project in the IDE

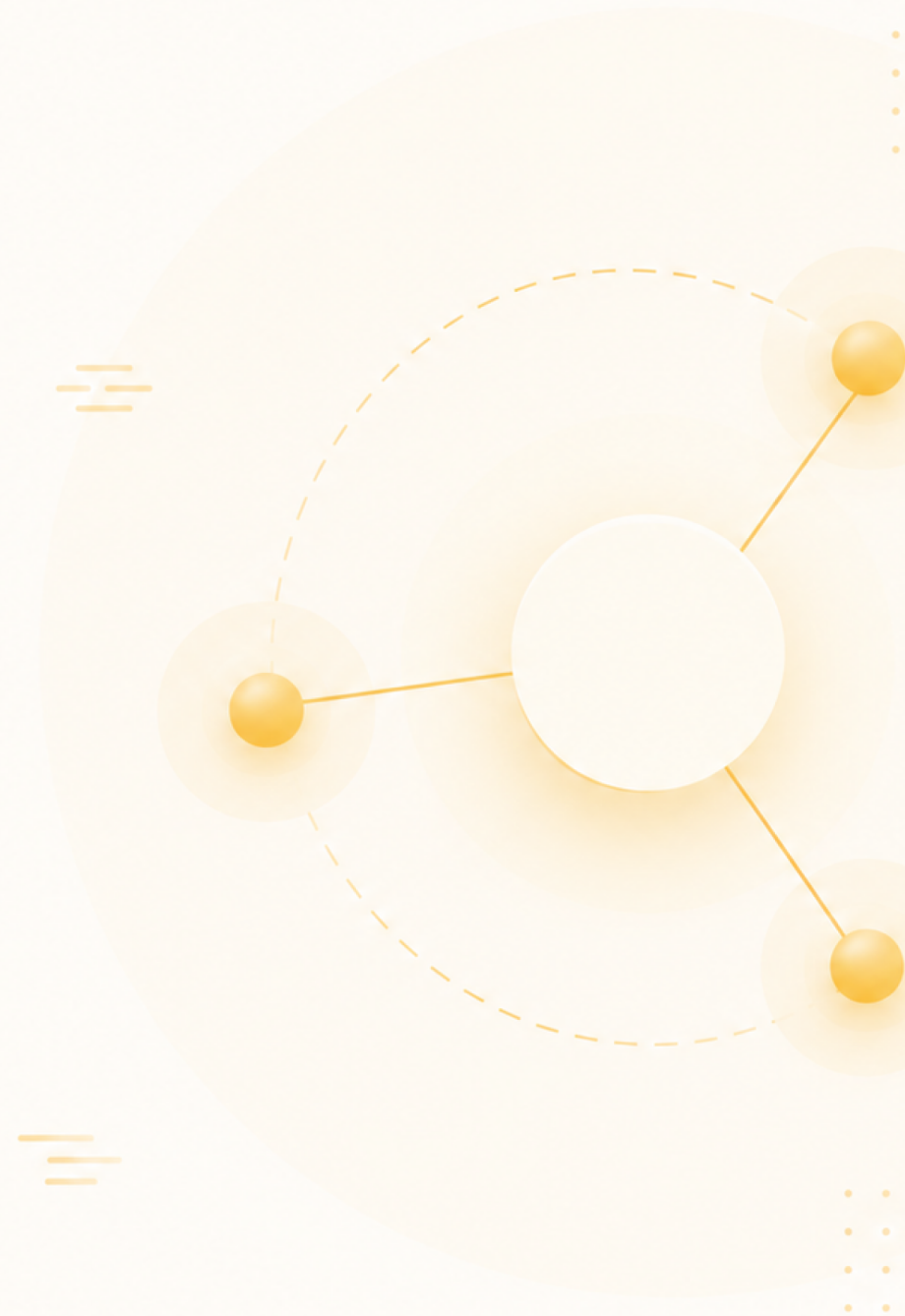
A team forks a community CloudKitchen ordering platform, downloads it as a ZIP, and opens it in VS Code. They use C0 to understand how the existing cart system works and review it for issues. They then switch to CORA in Plan mode and ask it to add a restaurant owner dashboard with order management and analytics. CORA reads the existing code, plans the new feature to match established patterns, and implements it across the codebase.

Example CORA prompt

Add a restaurant owner dashboard. Include order management with status updates (pending, preparing, ready, delivered), daily revenue analytics, and a menu management section. Match the existing component and routing patterns.

CHAPTER 5

Use Cases



5. Use Cases

Rapidly prototyping new product ideas

Trigger: A team wants to review, test, and share a running application without days of environment setup and scaffolding.

Result: A working prototype built on real code that proves out a direction and can be continued rather than discarded.

- Describe the product in Conversational Mode include the core pages, user roles, and key features
- Review the running application in the live preview
- Go live and share the URL with stakeholders for feedback

Example prompt

Build a task management app with a Kanban board, user authentication, team workspaces, and email notifications. The board should support drag-and-drop between columns.

Converting Figma designs into working applications

Trigger: A design team has an approved Figma file that needs to become a real application.

Result: A complete application generated with high fidelity using React components, Tailwind CSS, and the exact colours, typography, and layouts from the source file.

- Connect Figma using a Personal Access Token
- Paste the Figma frame or file URL and click Start Import
- BUILD reads all components, layouts, styles, colours, and typography
- Refine with follow-up prompts and go live

Example prompt after import

Add form validation to all input fields, connect the dashboard charts to the mock API, and add a loading state to each data panel.

Generating internal tools and dashboards

Trigger: A team needs an internal tool, dashboard, or admin panel without taking engineering time away from product development.

Result: A functional internal tool produced quickly, deployed to a live URL, and ready for the team to use without custom infrastructure.

- Describe the internal tool in Conversational Mode specify the data it displays, the actions users can take, and any role-based access requirements
- Generate, review in live preview, and go live without dedicated UI engineering time

Example prompt

Build an internal support ticket dashboard with ticket list, status filters (open, in-progress, resolved), priority labels, and an admin view showing all agents and their ticket counts.

Building and testing UI variations

Trigger: A team needs to compare multiple visual directions before committing to an implementation.

Result: A validated visual direction, exported to a fully functional application matching the approved design.

- Use Design Mode to generate multiple UI variations from a single description
- Compare them side by side and refine with follow-up prompts
- Export the approved design to code and go live

Example prompt

Design a SaaS landing page. Two variations: one minimal and light with serif headings, one bold and dark with large hero typography. Both should include a pricing section and a feature comparison table.

CHAPTER 6

Tips, Patterns, & Troubleshooting



6. Tips, Patterns and Troubleshooting

Tips and patterns

- Be specific about what the app does, who uses it, and how it should look. Describe the pages and screens, the key components and icons, the colour scheme and typography, and the user roles and flows. The more detail in the initial prompt, the closer the first output is to what you have in mind.
- Build incrementally through follow-up prompts rather than trying to describe everything upfront. Add features progressively and validate each addition in the live preview before moving on.
- Use the + button to attach files, photos, and screenshots. Attach a PRD to build from a specification, attach a UI screenshot to clone or reference a design, or take a screenshot directly from BUILD to give context on the current state.
- All generated code belongs to you. Download a ZIP at any time to push to GitHub or deploy on your own infrastructure independently of CodeMate Cloud.

Troubleshooting

Issue	Cause	Solution
BUILD is not generating what you asked for	The prompt is too broad or missing key details.	Refine the prompt to be more specific. Describe the app type, the core pages and user roles, the visual style and key components, and any technical requirements. Use iterative building: start broadly, validate in the live preview, and add detail through follow-up prompts.
The app is not going live after clicking Go Live	A connection issue, a subdomain conflict, or errors in the generated app.	Check your internet connection and try again. Ensure the chosen subdomain does not conflict with an existing project. If the app showed errors in the live preview, fix them through follow-up prompts before going live publishing a broken build results in a non-functional URL.

Figma import is not reading the design correctly

The PAT does not have all required scopes, or Auto Layout is not used in the Figma file.

Regenerate the PAT in Figma with all scopes enabled. In your Figma file, apply Auto Layout to all frames and name layers clearly so BUILD can accurately read the structure and components.

7. Resources and Reference

Where to go next

Product documentation: **docs.codemate.ai**

CodeMate website: **codemate.ai**

For issues beyond this guide, see the CodeMate BUILD FAQ and Troubleshooting documents.

Quick checklist

Use this before considering your BUILD project production-ready.

- ✓ Chose the right starting mode: Conversational, Design, or Figma Import
- ✓ Used the + button to attach any reference files, designs, or screenshots
- ✓ Validated the app thoroughly in the live preview before going live
- ✓ Clicked Preview site, then Go Live, and confirmed the Live URL works
- ✓ Set a custom subdomain via Publishing & Domains, or transferred to Netlify for a custom domain or CI/CD
- ✓ Forked or downloaded the project if you plan to continue in the IDE with C0 and CORA