

What is MCP?

Model Context Protocol (MCP)

The Model Context Protocol (MCP) is an open-source standard that enables AI applications to connect with external systems. Think of it as a universal translator that allows AI, like chatbots and other AI agents, to communicate with various data sources, tools, and applications.

What problem does MCP solve?

AI Data is real time, in multiple formats

Data is Complex
ie: REST and GraphQL resolution is complicated

API data and engineering logic and direction of data required compute and complex engineering systems integration, is often out of sync

Scaling and Mirroring of streaming data with versioning and location is difficult

Privacy and Security is difficult to manage

Brittle Single Point of Failure Systems require flexible data managers. Flexible data de risks vulnerability.

What are the advantages?

It is super simple to setup

Each Organization sets up Data Modeling tailored to their own specific data shape and security protocols

It is platform agnostic and can deploy on prem, supplemented by connected data shaped to BAU tools: CRM, ERP, IEP, CMS, DAM

Integration can be omni channel with tools, customized security and role based access that is Scoped to the organizations' logic and Business Rule

What is JSON RPC?

JSON-RPC, which stands for JavaScript Object Notation Remote Procedure Call, is a lightweight and stateless protocol that allows one program to request a service or procedure from another program on a remote computer as if it were a local call. It uses JSON (JavaScript Object Notation) to format the data for these requests and responses. The protocol is designed to be simple, language-independent, and transport-agnostic, meaning it can be used with various programming languages and over different network protocols like HTTP or TCP. A client sends a request object containing the method name to be invoked and any necessary parameters, and the server returns a response object with the result or an error.

How does it compare to REST

Aspect	JSON RPC	REST
Paradigm	Procedure-oriented	Resource-oriented
Operations	Defined by methods	Standard HTTP verbs (GET, POST, PUT, DELETE)
Flexibility	Less flexible, tightly coupled	More flexible, loosely coupled
Use Cases	Function calls, blockchain Streaming Data	CRUD operations, data retrieval

Facets/Filters

Collections

Products

Tags

Users

Events

Inputs

Schema/Type

Text

Integer

Image

Email

Phone

Link

Boolean

Rich Text

Multi Reference

Reference

Agents
'the brains/persona'

Gemini

Anthropic/Claude

Grok

Open AI

Tools

Inputs

APIs Endpoints

Agents/Tasks/Logic

User Eligibility Functions

Form Submission/
Redirection

Locale and Language
Detection

Privacy Settings

API Key Conditions

MCP
'asset management for AI'

Host

Client

Knowledge Base

Training
Instructions for AI

Xano

API First Backend

APIs

<https://docs.xano.com/building/logic/apis>

AI Agents

<https://docs.xano.com/ai-tools/agents>

MCP Servers/with SOC2 compliance

<https://docs.xano.com/ai-tools/xano-mcp-server>

Build with MCP

<https://docs.xano.com/ai-tools/mcp-builder>

Custom Functions that work Cloud to Cloud SAAS to SAAS bridge

<https://docs.xano.com/the-function-stack/functions/custom-functions>

Background Tasks

that resolve Rate Limits that are fee upcharge SAAS strategies

<https://docs.xano.com/the-function-stack/building-with-visual-development/background-tasks>

Triggers

especially important if your org doesn't utilize GA4 Enhanced Measurements

<https://docs.xano.com/the-function-stack/building-with-visual-development/triggers>

Middleware-SINGLE FEE

Unlimited APIs paired with Middleware that is Multi Tenant across DNS and Top Level/Middle 'hreflang' naming globalization is 'Priceless'

<https://docs.xano.com/the-function-stack/building-with-visual-development/middleware>

Building APIs in Xano - Xano DocumentationLearn how to build APIs with no code, with code, or with AI, all in Xano

Spec-driven design, or more formally spec-driven development (SDD), is an emerging approach where structured specifications are created and maintained as the main source of truth for software projects—often with help from AI agents. Instead of starting with code, teams first create detailed, behavior-focused “specs” that describe functionality, interfaces, and requirements. These specs guide both human and AI-assisted development, ensuring consistency and clarity throughout the software lifecycle.

/constitution

/specify

/clarify

/plan

/task

/analyze

Phase	Focus	Key Activities
0-to-1 Development (“Greenfield”)	Generate from scratch	Start with high-level requirements Generate specifications Plan implementation steps Build production-ready applications
Creative Exploration	Parallel implementations	Explore diverse solutions Support multiple technology stacks & architectures Experiment with UX patterns
Iterative Enhancement (“Brown-“)	Brownfield modernization	Add features iteratively Modernize legacy systems Adapt processes

Sample Work Flow

Business Goal is to convert Single Tenant Data that is not visible to Machine Index/Node Search to JSON that can be found by SERP engines.

MCPs used: Webflow, Netlify, Github

APIs and Workers endpoints
(‘Headless Functions’)
Xano/Cloudflare Data Modeling Logic

Security and Entitlement: oAuth/Auth0 tokens

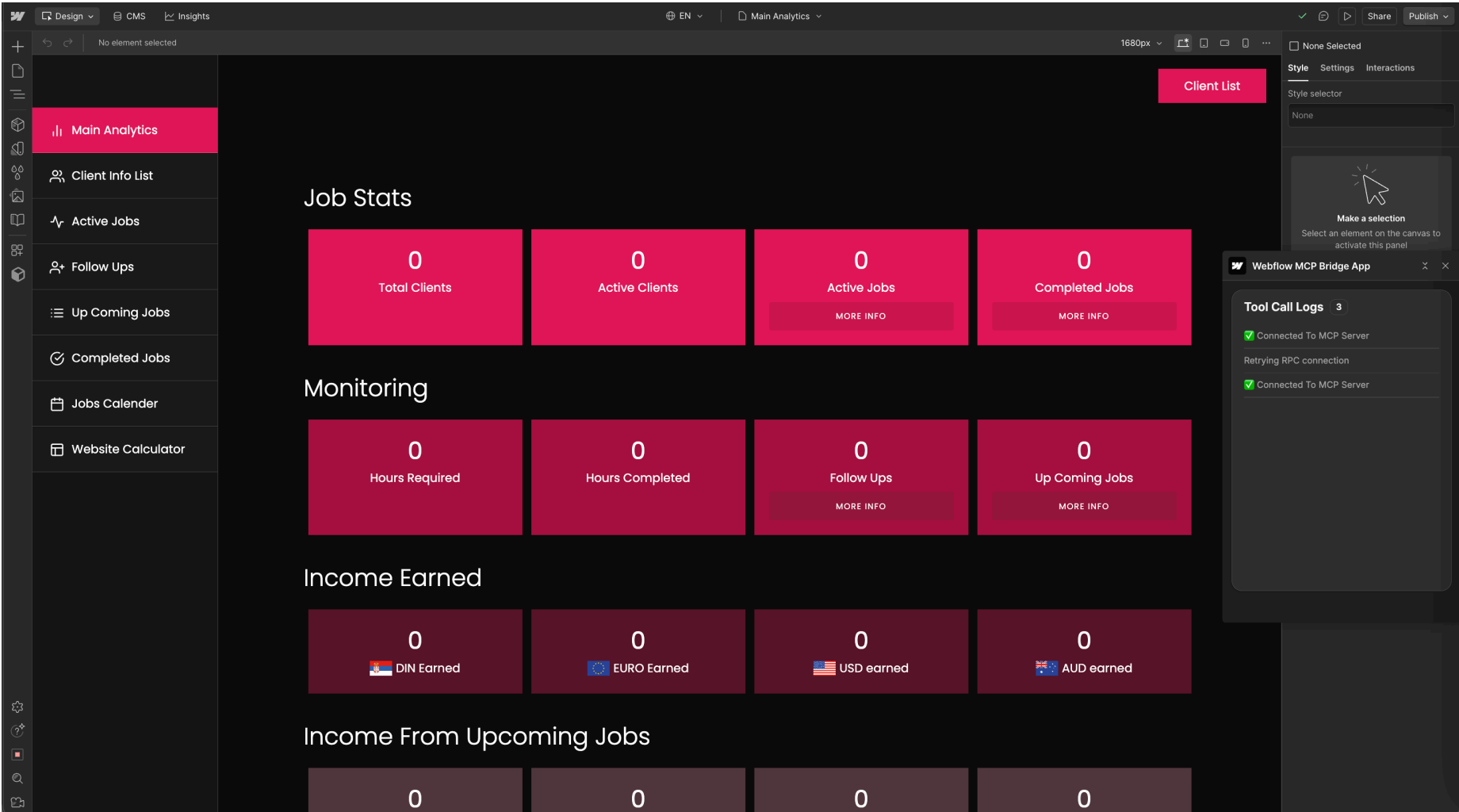
Data Portability and Encryption Endpoints Xano

Time Tracker Configurator with MCP Build

<https://taskmanager-sample.netlify.app/>

[Github Public Source Code](#)

[Webflow Clonable \(no edits made, as is\)](#)
[Use as baseline for Agentic Prompts](#)



Agentic PRDs

Product Requirements Document

Open AI / Shopify	An exploration of OpenAI's API-backed agentic commerce and a pending Shopify Chatbot, demonstrating a PIM build with MCP/Xano. This work has direct applications to anticipated projects at HP.com.	https://manus.im/share/nO6PaQaajcoRZflj6au4MC?replay=1 OpenAI Commerce Specs
Twitch API/VIBE	A 24-hour challenge to duplicate a codebase linking HP.com to a Twitch game stream. The process involved prototyping in Webflow, working with the Twitch API, and deploying to GitHub Pages to Shopify page.	https://manus.im/share/yHUQC1FrjWhCNOcySp5BgK?replay=1 https://persephonepunch.github.io/omen-micro/
Shopify Klaviyo GraphQL API	Migration Data	https://manus.im/share/x7bAaWpxapfNBS3omqredL?replay=1
Shopify API Data Calendar	Deprecation Calendar and Recommendation Strategy	https://claude.ai/share/9daa9ffd-a003-4198-9792-4eb43e5b3a3d
Xano / Functions	PIM / Agentic API endpoints	https://chatgpt.com/share/68e6c66d-e2d4-8012-a436-10872e00b343
Big Query / SEM Rush Builder	Replace Fee Based SEO SAAS with open source API data	https://claude.ai/share/0d3137e1-c7e1-418d-9d68-617e270185a4
Shopify to Unity	A workflow starting from flat design in Illustrator and After Effects, moving through to 3D with Cinema 4D and Adobe Dimension, and finally to a gaming environment in Unity. This example also highlights the potential of Shopify Web Components for expanding agentic design tools.	https://claude.ai/share/0d3137e1-c7e1-418d-9d68-617e270185a4
Media Storage	Design Library and Media Hosting A demonstration of publishing and hosting video assets with a responsive modal view. This involved using Replit /Media to Shopify page.	https://manus.im/share/7E98vVBaGWFLadKJyreq7K?replay=1 https://webcomponents.shopify.dev/playground?view=editor&pre-set=product-cards-discover
Front End Dev	Real Time XD/Figma to Design Components Build, Fonts, Images, Docs Web C to Shopify CDN	https://headlessshopify.netlify.app/
LLM Shopify / 11ty / Web C	Web C Static Build outs with 11ty/Shopify Product Feed/Filter/Prompts	https://manus.im/share/e5BIZ45fYDS3Hu5Ey8OZis?replay=1 https://filters-hx.netlify.app/ https://hyperx-catalog.netlify.app/
LLM Prompt	Shopify Web Components LLM with inline CSS example	https://webcomponents.shopify.dev/lms.txt
Agentic Scaffold	Shopify Web Components and 11ty Generative AI Scaffold	https://github.com/persephonepunch/11ty-shopify-site.git
Hydrogen Scaffold Shopify	Shopify Hydrogen/React Deployed to Netlify	https://hydronettemplate.netlify.app/ Netlify/Shopify Reference https://docs.netlify.com/build/frameworks/framework-setup-guides/hydrogen/

Business Objective

Convert Structured Collection Data to JSON / Agentic Legible Objects
Define Schema using Low Code to Developer API endpoints
Setup Data Design Guidelines that are high performing high fidelity

Opportunity

Streamline Process Management
Introduce Private Keys/Security
Create Individual Dashboards
Use AI to summary projects and document store
Use Static Files paired with Open Source Publishing to structure asset management
Use Familiar Tools
Introduce Design Tools: Figma, Prototype, Image, Animation, Gaming

Procedure

- 1) Setup Existing Project Tracking Format
- 2) Connect to Cursor or Claude
- 3) Gather Requirements
- 4) Deploy to Github/Vercel/Netlify
- 5) Iterate with CMS
- 6) Version and Optimize Tools/Collections and Utility

References

Webflow MCP

<https://developers.webflow.com/data/docs/ai-tools>

Shopify MCP

<https://shopify.dev/docs/apps/build/devmcp>

Claude Desktop

<https://claude.ai/download>

Cursor Install

<https://cursor.com/download>

Webflow LLM TXT Knowledge Base

<https://developers.webflow.com/llms.txt>

Shopify LLM TXT Knowledge Base

<https://webcomponents.shopify.dev/llms.txt>

Figma MCP to Webflow MCP

https://youtu.be/wQW341X_Y5k?si=LRSbVBcTI5TQ3Q0B

Cursor Deploy Task Manager

Webflow 11ty Export

Webflow Cursor MCP

Netlify Cursor Deploy

Netlify MCP

Github MCP

Auth0 Entitlement

oAuth Entitlement

Better Auth API Gating

Slack Notifications

Webhooks

Server Actions

Xano Functions (Agent Builders plus Docker / Kubernetes Data Scale/Transport)

Open AI Agent Builders

Data / Design / AEO managers

Cloudflare Workers

Github Pages

Cloudflare Pages

Automation and Webhooks

Core Conversion Goals to move 'Typed' nested data to File System managed real time NLP/MCP accessed data endpoints

React Component Definitions Server to Client Methods to manage Code Load and CI CD Automation

<https://youtu.be/S5tjBqzs31w?si=00n2-NGsJu7gfevO>

SSR (default) for Next

Server-Side Rendering (SSR)

Default Renders on Server Hydrate the Browser

Renders on the Server: The server generates the full HTML for a page at the time of the request.

SEO Friendly: Because the content is already in the HTML when it's sent to the browser, search engines can easily crawl and index it.

Faster Initial Load: Users see the page content more quickly because the browser receives a fully rendered HTML file.

Ideal for Dynamic Data: It's well-suited for pages that need to display data that changes often, like e-commerce sites or user-specific pages.

ISR

Incremental Static Regeneration

Use Case Form input is updated.

Dynamic 'Blog' or 'Product Feed' is pushed

CI CD Server Events

Github/Vercel Server Actions

'on change'/revalidate

Next JS Revalidate Path

Next JS Revalidate Tag

On change url/tag can also fire:

GTM 'Tag'

CSR when Browser is needed ie: onclick, animation, form submit

Client-Side Rendering (CSR)

"use client";

Webflow 'State' Animations are Client Rendered

Renders in the Browser: The browser receives a minimal HTML file and a JavaScript bundle, and the JavaScript is responsible for rendering the content.

Rich Interactivity: After the initial load, navigating between pages can feel very fast and fluid, making it great for interactive applications like dashboards.

Lower Server Load: The server's main job is to send the initial files, and the client's browser handles the rendering, which can reduce server costs.

Slower Initial Load and SEO Challenges: The initial page load can be slower because the browser has to download and execute JavaScript, and it can be less SEO-friendly out

PPR

Partial Pre Rendering

cached cookies/session data
that is temporary and stores
Dynamic Render based on load

Hybrid Rendering on a Single Page: It blends the instant load speed of static generation with the real-time data capabilities of server-side rendering.

Instant Static Shell: Users get an immediate, non-interactive view of the page, which dramatically improves the perceived performance and Largest Contentful Paint (LCP).

No New APIs to Learn: PPR is designed to work with your existing Next.js components. You simply use React's <Suspense> component to wrap the dynamic parts of your page. Next.js automatically understands that these wrapped components should be streamed in.

Improves User Experience: It eliminates the "all or nothing" trade-off between SSG and SSR. You no longer have to choose between a fast but static page or a dynamic but slower-loading page. With PPR, you get both.

SSG

Static Site Generation (SSG)

Renders at Build Time: The HTML for each page is generated once when you build your application for production, not when a user makes a request.

Extremely Fast Performance: Since the pages are pre-built as static HTML files, they can be served directly from a Content Delivery Network (CDN), resulting in incredibly fast load times.

Excellent for SEO: Search engines can easily crawl and index the pre-rendered HTML files, making SSG a great choice for SEO-focused content like blogs, marketing sites, and documentation.

Ideal for Static Content: It is best suited for content that does not change frequently, such as blog posts, documentation pages, marketing landing pages, and product catalogs.

Webflow export is STATIC bundle
Default React serving on Client/
Static CSS, JS, HTML

Shopify Theme Liquid
is CSR and Remix APIs are SSR

PIM/GTM Data layer and Flows can
funnel data into CSR using SSR
GraphQL Server Modeling

Next is both
Remix is only SSR

Webflow Devlink can export/import
React Components to async data and
internally to CSR build outs

Webflow Cursor MCP

Preview and Schema Logic

<https://projecttracker-eee266.webflow.io/> Relational Data is converted to Mark Down/Liquid-JSON API endpoints

CMS Collections

Project Managements 13 Items

Project Name

Published

Status

Created

Modified

Archiscale

Oct 7, 2025 10:57 PM

Published

Sep 24, 2021 7:08 PM

Sep 24, 2021 7:08 PM

ABC Auto

Oct 7, 2025 10:57 PM

Published

Sep 24, 2021 7:02 PM

Sep 24, 2021 7:03 PM

Art Pres

Oct 7, 2025 10:57 PM

Published

Aug 3, 2021 4:39 PM

Sep 24, 2021 7:10 PM

Delerius Form Setup

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 8:28 PM

Sep 24, 2021 7:12 PM

Daniels Ecommerce Websites

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 8:14 PM

Sep 24, 2021 7:14 PM

Jewelry Store Ecommerce

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:57 PM

Sep 24, 2021 7:15 PM

Lampone

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:44 PM

Sep 24, 2021 7:29 PM

Instagrab

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:37 PM

Sep 24, 2021 7:37 PM

Shopify Project

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:31 PM

Sep 24, 2021 7:20 PM

Healthrix Website

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:28 PM

Sep 24, 2021 7:22 PM

Healthrix Software

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:25 PM

Sep 24, 2021 7:24 PM

Realestate Updates

Oct 7, 2025 10:57 PM

Published

Aug 1, 2021 7:08 PM

Sep 24, 2021 7:26 PM

Store custom pages CRM

Preview

Code

Blame

39 lines (38 loc) · 1.87 KB

Raw

Download

Copy

Share

```
1 ---
2 f_client-name: Mike Rogers
3 f_client-address: 27 Nariel Street, Albion Queensland 4010, Australia
4 f_client-tax-id: A85299328002
5 f_invoice-generate-date: '2021-09-22T00:00:00.000Z'
6 f_invoice-validation-date: '2021-09-30T00:00:00.000Z'
7 f_invoice-number: '01'
8 f_hours-required: 66
9 f_due-date: '2021-10-04T15:00:00.000Z'
10 f_client-email: admin@abcautohub.com.au
11 f_project-cost: '6930'
12 f_currency: AUD
13 f_client-active: true
14 f_invoice-received: true
15 f_payment-received: true
16 f_client-notes: Client likes mockups, ready to move to development stage.
17 f_payment-stages: '3'
18 f_current-payment-stage: 2
19 f_payment-amount-for-current-stage: '2310'
20 f_follow-up: false
21 f_upcoming-job: false
22 f_currently-being-worked-on: true
23 f_job-completed: false
24 f_select-language: English
25 title: ABC Auto
26 slug: abc-auto
27 created-on: '2021-09-24T17:02:44.548Z'
28 updated-on: '2021-09-24T17:03:27.392Z'
29 published-on: '2025-10-08T01:36:45.229Z'
30 layout: '[project-management].html'
31 tags: project-management
32 ---
```

Postgres / CSV file formats

Machine VIBE legible Markdown Files

‘Type Safe’ Schema Modeling
Typescripting Data

CMS Collections

Project Management Settings

Project Managements 13 Items

Collection fields

+ Add New Field

Ecommerce

Basic Info

Custom fields

T Project Name (Plain text)

T Client Name (Plain text)

T Client Address (Plain text)

T Client Tax ID (Plain text)

Invoice Generate Date (Date/Time)

Invoice Validation Date (Date/Time)

T Invoice Number (Plain text)

Project Requirements (Rich text)

Hours Required (Number)

Due Date (Date/Time)

Client Email (Email)

Attachments (File)

T Project Cost (Plain text)

Currency (Option)

Client Active (Switch)

Invoice Received (Switch)

Payment Received (Switch)

Client Notes (Rich text)

Payment Stages (Option)

Current Payment Stage (Number)

T Payment Amount for current stage (Plain text)

Follow Up? (Switch)

Upcoming Job (Switch)

Currently Being Worked On (Switch)

Job Completed? (Switch)

Select Language (Option)

+ Add Field

Basic Info

Project Name *

Slug *

webflow.com/project-management/example

Custom fields

Client Name *

Client Address

Client Tax ID

Invoice Generate Date

Invoice Validation Date

Invoice Number

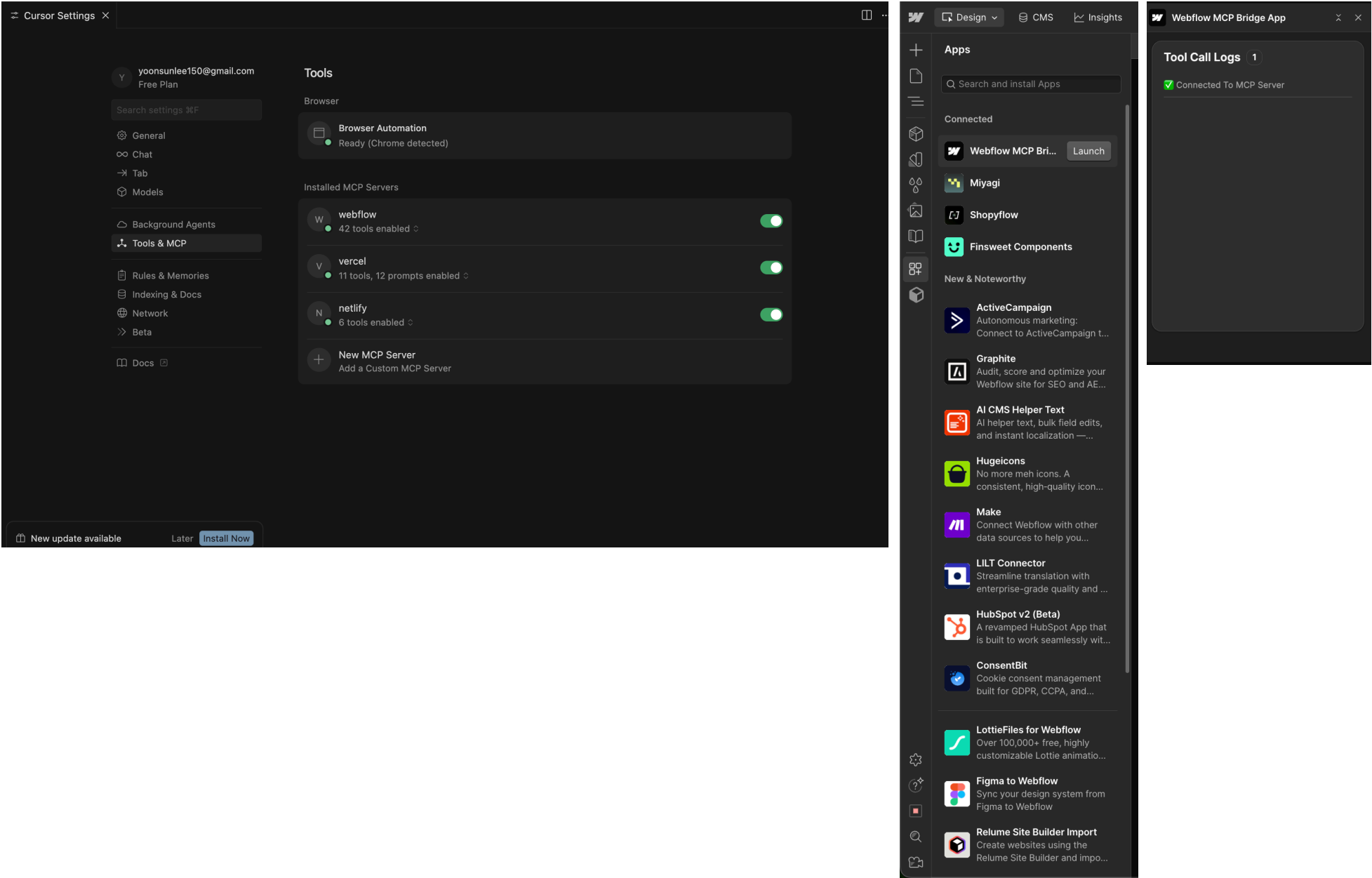
Project Requirements

Hours Required *

Due Date *

Webflow Cursor MCP

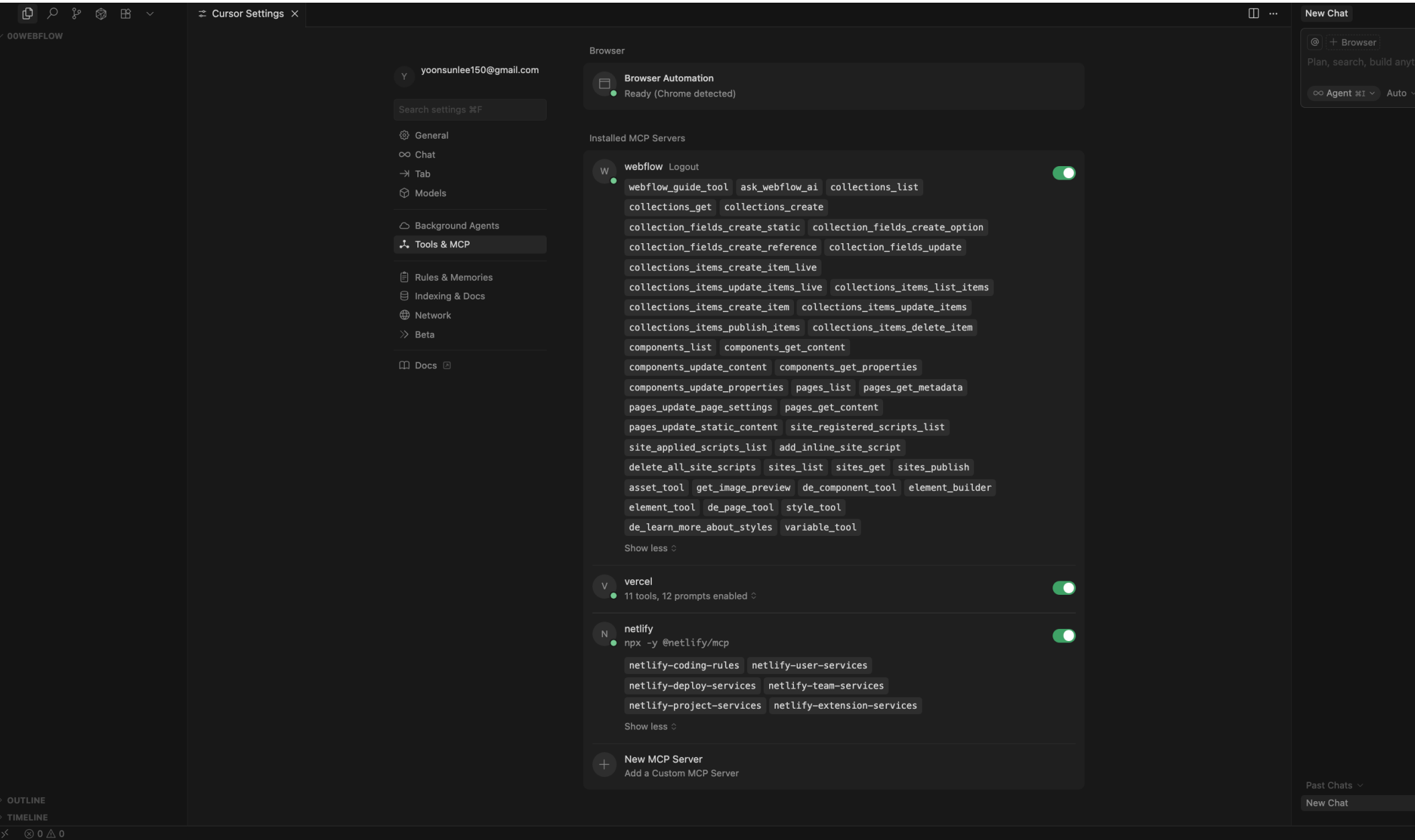
Preview and Schema Logic. Installation link <https://developers.webflow.com/data/docs/ai-tools>



Chained MCP Servers/WebFlow App Bridge

Data Synced Collections, via webhooks, rpc managers: Function Events: GTM/Xano/Lamdas/Agentic Builders, can be chained with Build and Deploy tools linked by connected AI enabled data rules. Collections can be Users, Orders, PIM - Product Feed (GTM Data Layer is a 'live data system').

Functions / Flows are created by 'choose your adventure' natural language goal setting.





CI CD Deploy to Github / Github MCP / Server Actions / SEO/Accessibility and AUTH exports with APP. Export to standard BAU CMS/CRM/DAM/Ecommerce workflows

Exported templates and themes can be hooked to webhook UI / Asset Modeling / Optimization.

nexus

ExtensionUniversityTurn My Figma^{NEW}



App Older Version

 → 

Webflow to Shopify

Design your project in Webflow and run your e-commerce with Shopify, in few simple steps.



Getting Started

 → 

Webflow to WordPress

Enjoy a new limitless solution to convert a Webflow project into a real WordPress Theme without code.



Getting Started

 → 

Webflow to Jamstack

Use your Webflow project as a Jamstack powered website, without writing a single line of code.

Getting Started

 → 

Webflow to Ghost

Design meets functionality. Build in Webflow and manage a successful blog with Ghost.

Getting Started

CI CD/CDN/Deploy-Github as the Scaling Data Agent - SEO optimized Content Flows

Continuous Integration Version Control with MCP Code Optimization. Markdown AI Enabled Real Time Data Sets Scale and update with Github Actions MCP
Versioned Data Sets with time stamp and Headless CMS/Decap. API backed MCP Public-Private Event Hooks with Ledger/Version control

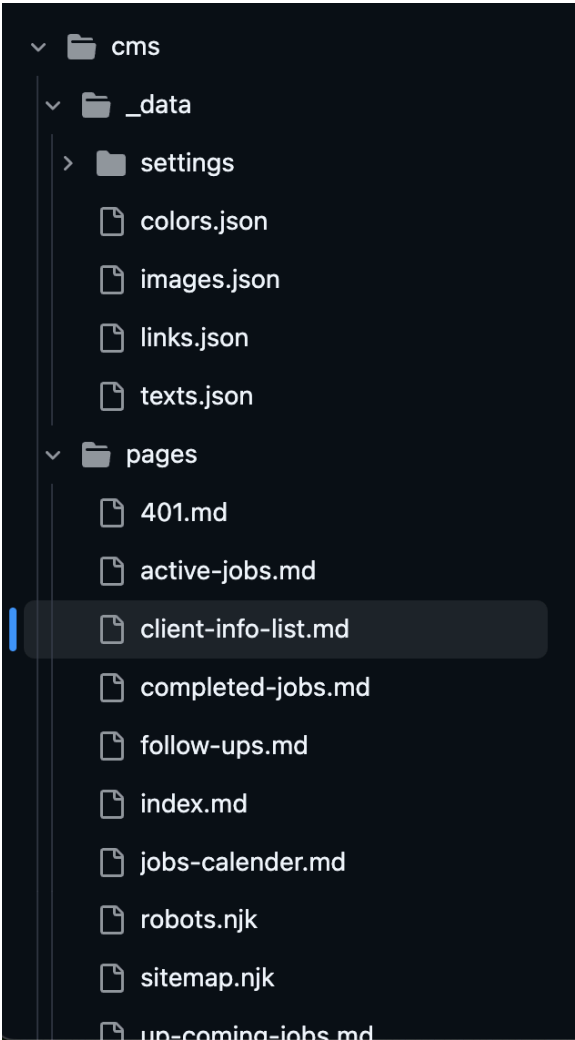
Shopify and Github Pages are both Ruby on Rails Templates and share Liquid/Jekyll Syntax. Github Data and MCP Scales without EGRESS fee and is unlimited.

Github Public

<https://github.com/persephonepunch/taskrunner.git>

Github Private Mirror

<https://github.com/persephonepunch/tasks-main>



Preview Code Blame 14 lines (12 loc) · 254 Bytes								
title	permalink	layout	slug	tags	seo			
Client Info List	{{ page.fileSlug }}/index.html	client-info-list.html	client-info-list	pages	noindex	title	og:title	twitter:title
					false	Client Info List	Client Info List	Client Info List

```
1 ---
2 title: Client Info List
3 permalink: '{{ page.fileSlug }}/index.html'
4 layout: client-info-list.html
5 slug: client-info-list
6 tags: pages
7 seo:
8   noindex: false
9   title: Client Info List
10  og:title: Client Info List
11  twitter:title: Client Info List
12 ---
```

Markdown (Liquid/Nunjucks)
Machine Legible Dynamic Files

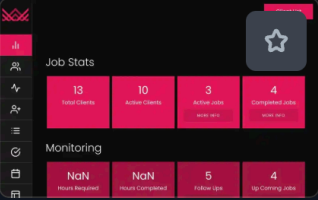
Cursor Netlify Functions enabled File Stacks
JSON LD and SEO optimized Content Management


YAML front matter is SEO/AEO and AI Legible and Function enabled

Netlify MCP / Entitlement / Private Dashboard / Teams-RBAC SSO Admin Tools

<https://developers.netlify.com/guides/write-mcps-on-netlify/>

<https://taskmanager-sample.netlify.app/>




taskmanager-sample 

● **taskmanager-sample.netlify.app**

Deploys from **GitHub** with **11ty** **Eleventy**.

Published at 9:53 AM.

[Quick setup](#) [Add database](#) [Customize](#) 

Slack notifications

Configure Slack subscriptions for project activity, including deploys, form submissions, collaboration, and configuration changes

[Learn more about Slack notifications in the docs](#)


Workspace installation

Connect to your Slack workspace to subscribe to real-time notifications about activity from your Netlify team and projects.

Depending on how your Slack workspace admins have configured app permissions, you may need to get installation approved.

[Learn more about the Netlify App for Slack in the docs](#)

[Select workspace](#)



Install authentication provider

Provider

- ☒ GitHub
- ☐ Bitbucket
- ☐ Google
- ☐ GitLab

Secret

[Install](#) [Cancel](#)

Auth0

Auth0 is a comprehensive and universal authentication and authorization platform that helps you manage application access.

[Global extension settings](#)


[Extension documentation](#)

Site tenants

Connect Auth0 tenants to one or more site deploy contexts and automatically generate applications and APIs. Once you're done, you can apply matching environment variables to your site.

[Add a tenant](#)

yoonsun lee's Auth0 configuration

de  d [Remove](#)

Domain [Link an Auth0 tenant](#)

To connect your Netlify site to Auth0, an Auth0 tenant needs to be connected to your Netlify account. Tenants are linked to your Netlify user (yoonsun lee) and can be used across all sites in your team once connected.

[Learn more about Auth0 tenants](#)

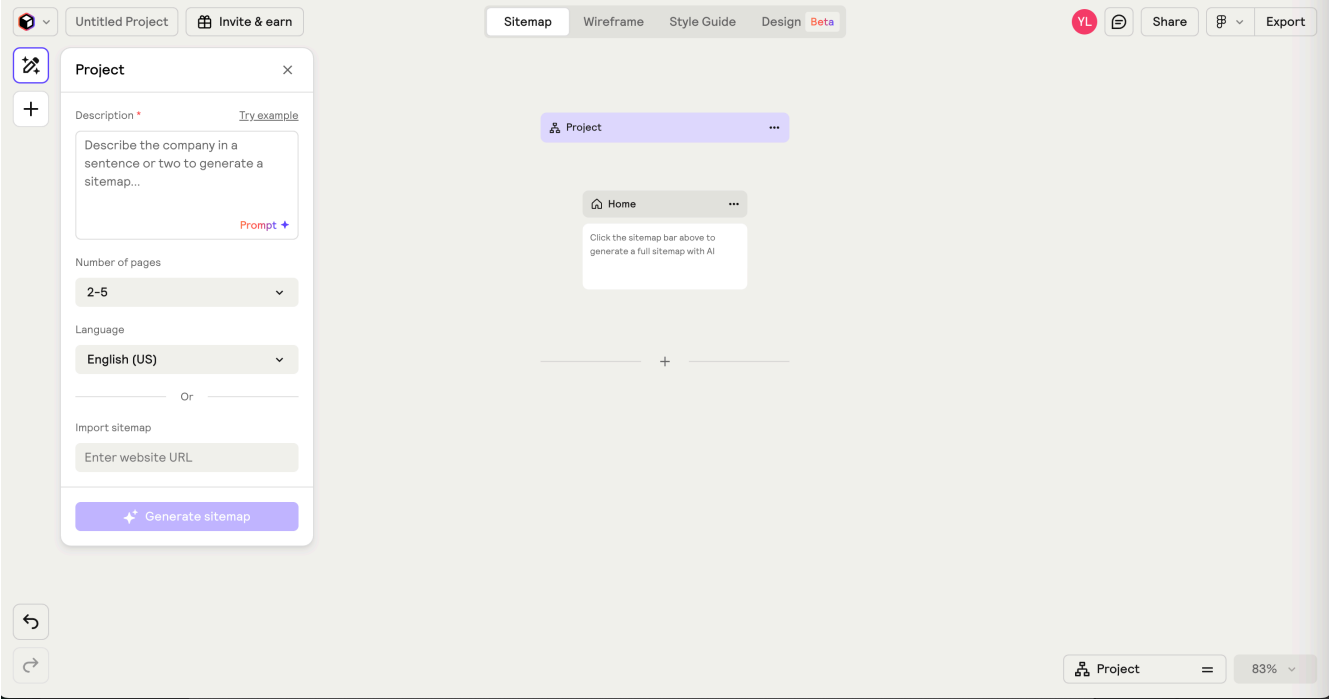
[Link an Auth0 tenant](#)

Wireframe, Figma, React, Webflow

Design Resources

Relume / Figma / React

<https://www.relume.io/?r=0>

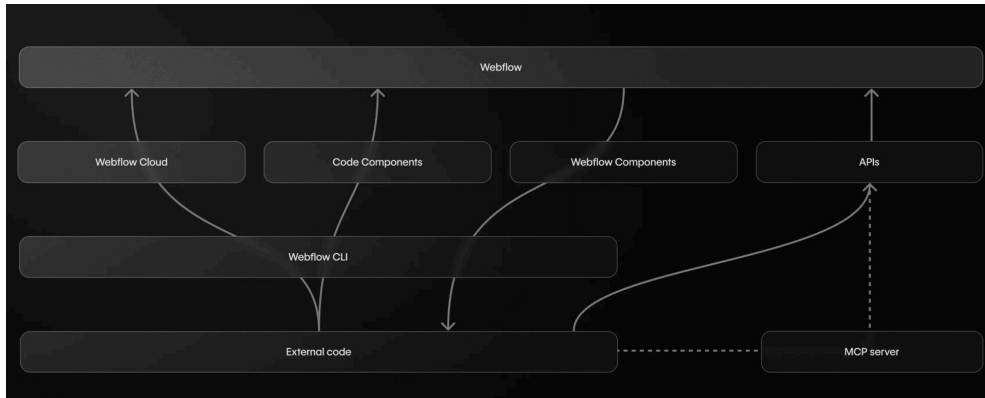


App Build Getting Started:

Introduction

Webflow Cloud is also Cloudflare / Cloudflare Workers and KV Files is included with Compile and Assets Optimization

<https://developers.webflow.com>



App capabilities



Data Clients

Harness your site's data

Sync CMS content, form submissions, and translations between Webflow and your favorite tools. Push blog posts to social media, send form responses to your CRM, and keep content in sync across platforms using Webflow's Data APIs.



Designer Extensions

Supercharge the design workflow

Build extensions that automate complex design tasks in Webflow. Help users focus on creativity by streamlining repetitive work - from importing and transforming design assets to applying sophisticated styles and managing content updates across pages.



Hybrid Apps

Create seamless experiences

Combine the power of Data Clients and Designer Extensions to create comprehensive solutions. Build apps that seamlessly integrate content management through APIs while providing direct design controls in the Webflow Designer - giving users a unified experience to manage their entire workflow.

Project Structure / Webflow Cloud is Cloudflare Cloud
Cloudflare also DNS Manages Shopify / One Trust

Project Structure

```
.
├── data-client/                                # Backend server
│   ├── app/
│   │   ├── api/                                # API Routes
│   │   │   ├── auth/                          # Auth endpoints
│   │   │   └── custom-code/                   # Custom code endpoints
│   │   ├── lib/                               # Server utilities
│   │   └── controllers/                       # Logic for handling requests and responses using
│   │       └── db/                            # Database
│   ├── .env.example                           # Environment template
│   └── package.json
├── designer-extension/                        # Frontend app
│   ├── src/
│   │   ├── components/                       # React components
│   │   ├── hooks/                           # Custom hooks
│   │   ├── services/                         # API services/logic
│   │   ├── types/                           # TypeScript types
│   │   └── App.tsx                           # Main app component
│   ├── .env.development                      # Dev environment variables
│   ├── package.json
│   └── package.json                          # Root package.json
```

Additional Resources

- [OAuth 2.0 Implementation Guide](#)
- [Hybrid App Authentication](#)
- [Available API Scopes](#)

Main Dashboard Pages

The system has several dashboard views defined in the **`Pages`** collection:

- Main Analytics** (**`index`**)
- Client Info List**
- Active Jobs**
- Follow Ups**
- Up Coming Jobs**
- Completed Jobs**
- Jobs Calendar**
- Website Pricing**

Project Management Schema

The primary data schema is the **`project_management`** collection with the following fields:

Basic Information

- title - Project Name (string, required)
- created-on - Created On (datetime)
- updated-on - Updated On (datetime)
- published-on - Published On (datetime)
- body - Project Requirements (markdown)

Client Information

- f_client-name - Client Name (string, required)
- f_client-address - Client Address (string)
- f_client-tax-id - Client Tax ID (string)
- f_client-email - Client Email (string, required)
- f_client-notes - Client Notes (markdown)
- f_client-active - Client Active (boolean, default: false)

Project Details

- f_hours-required - Hours Required (number, required)
- f_due-date - Due Date (datetime, required)
- f_project-cost - Project Cost (string)
- f_currency - Currency (select: USD, AUD, EU, DIN)
- f_attachments - Attachments (object with file URL)

Invoice Information

- f_invoice-generate-date - Invoice Generate Date (datetime)
- f_invoice-validation-date - Invoice Validation Date (datetime)
- f_invoice-number - Invoice Number (string)
- f_invoice-received - Invoice Received (boolean, default: false)
- f_payment-received - Payment Received (boolean, default: false)

Payment Structure

- f_payment-stages - Payment Stages (select: 1-16)
- f_current-payment-stage - Current Payment Stage (number)
- f_payment-amount-for-current-stage - Payment Amount for current stage (string)

Job Status Flags

- f_follow-up - Follow Up? (boolean, default: false)
- f_upcoming-job - Upcoming Job (boolean, default: false)
- f_currently-being-worked-on - Currently Being Worked On (boolean, default: false)
- f_job-completed - Job Completed? (boolean, default: false)

Other Settings

- f_select-language - Select Language (select: English, German, Serbian)
- f__noSearch - No Search (boolean, default: false)
- tags - Fixed value: "project-management"
- layout - Fixed value: "[project-management].html"
- seo - SEO metadata object

This schema is used with Netlify CMS (Decap CMS) for content management and utilizes the Eleventy (11ty) static site generator framework.

DATA as AI transformation JSON / Liquid endpoint

```
{
  "id": "68e5681febd6cca1cab1c637",
  "displayName": "Project Managements",
  "singularName": "Project Management",
  "slug": "project-management",
  "createdOn": "2021-08-01T16:57:05.642Z",
  "lastUpdated": "2025-10-07T19:21:07.909Z",
  "fields": [
    {
      "id": "c4dec44f230b476ccd35316e68039fe7",
      "type": "PlainText",
      "slug": "name",
      "displayName": "Project Name",
      "isRequired": true,
      "isEditable": true,
      "validations": {
        "maxLength": 256
      }
    },
    {
      "id": "4b5f7f370dc10e98e4b9796c151f7f5f",
      "type": "PlainText",
      "slug": "slug",
      "displayName": "Slug",
      "isRequired": true,
      "isEditable": true,
      "validations": {
        "maxLength": 256,
        "pattern": {},
        "messages": {
          "pattern": "Must be alphanumerical and not contain any spaces or special characters",
          "maxLength": "Must be less than 256 characters"
        }
      }
    },
    {
      "id": "be0b3b2a3e5aedad64ffe407fe0dac41",
      "type": "PlainText",
      "slug": "client-name",
      "displayName": "Client Name",
      "isRequired": true,
      "isEditable": true,
      "helpText": "Client Name",
      "validations": {
        "singleLine": true
      }
    },
    {
      "id": "61239497e18e5820ace67b2fd5524802",
      "type": "Email",
      "slug": "client-email",
      "displayName": "Client Email",
      "isRequired": true,
      "isEditable": true,
      "helpText": "Client Email"
    },
    {
      "id": "c2014fda341ce828e7573b1fc016506d",
      "type": "PlainText",
      "slug": "client-address",
      "displayName": "Client Address",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Client Address",
      "validations": {
        "singleLine": true
      }
    },
    {
      "id": "4ca530a447a9c260c0ccb2daae0b4201",
      "type": "PlainText",
      "slug": "client-tax-id",
      "displayName": "Client Tax ID",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Client Tax ID",
      "validations": {
        "singleLine": true
      }
    },
    {
      "id": "bf7b3171251fa1ea17394133f0ec0e85",
      "type": "RichText",
      "slug": "project-requirements",
      "displayName": "Project Requirements",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Project Requirements"
    },
    {
      "id": "7e1aefcc1edd24d838e0e72b71f2aa91",
      "type": "Number",
      "slug": "hours-required",
      "displayName": "Hours Required",
      "isRequired": true,
      "isEditable": true,
      "helpText": "Hours Required",
      "validations": {
        "format": "integer",
        "precision": 1,
        "allowNegative": false,
        "minValue": 1
      }
    },
    {
      "id": "295767bc65368af7a2d8e740948f3f98",
      "type": "DateTime",
      "slug": "due-date",
      "displayName": "Due Date",
      "isRequired": true,
      "isEditable": true,
      "helpText": "Due Date",
      "validations": {
        "format": "date-time"
      }
    },
    {
      "id": "8ce4cce27ee2719713a16c643803eb68",
      "type": "PlainText",
      "slug": "project-cost",
      "displayName": "Project Cost",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Project Cost",
      "validations": {
        "singleLine": true
      }
    },
    {
      "id": "76f68cd10a720cfd862ada371dbad3cc",
      "type": "Option",
      "slug": "currency",
      "displayName": "Currency",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Currency",
      "validations": {
        "options": [
          {
            "name": "USD", "id": "a5e-77d17034a4f217054dfe8b16b5db1"
          },
          {
            "name": "AUD", "id": "661741c55930bdc8cd-78c691983082c8"
          },
          {
            "name": "EU", "id": "81a6e0c029e37f3dc06cf-ce54fd18023"
          },
          {
            "name": "DIN", "id": "961779820ed701c86b3f-12b27367c498"
          }
        ]
      },
      "id": "24537ad5af473ada7b194a0588e7fc2f",
      "type": "Option",
      "slug": "payment-stages",
      "displayName": "Payment Stages",
      "isRequired": false,
      "isEditable": true,
      "helpText": "Payment Stages",
      "validations": {
        "options": [
          {
            "name": "1", "id": "a501115a9f7c-c0f7b0f52f76e6895223"
          },
          {
            "name": "2", "id": "787ca542b64ffc8edc-83b10a375224a0"
          },
          {
            "name": "3", "id": "18af56bb5d1ecb2e86db-870c955a3e02"
          },
          {
            "name": "4", "id": "ca4a31305fad0c1944edfc-3cfcc3e3b3"
          },
          {
            "name": "5", "id": "d6ba92f1e16e63fc493d-c8ed1e0f003c"
          },
          {
            "name": "6", "id": "3f464ec70b512e7daea-324783ca0acd0"
          },
          {
            "name": "7", "id": "f201c6e52fa51e-e7c2d43697d7309fa6"
          },
          {
            "name": "8", "id": "5cf31c9e-39032b8a714a44ff84d02b36"
          },
          {
            "name": "9", "id": "269d0c9a67dd0f9a7bfd-179f7b39b444"
          },
          {
            "name": "10", "id": "1eeeea1c07e0d4e7f-148142151dc9ffe0"
          },
          {
            "name": "11", "id": "6c2341c11f6f392d-534685c141d6df5f"
          },
          {
            "name": "12", "id": "8f091151158263db5d-6ba9e60b28cbe3"
          },
          {
            "name": "13", "id": "8dabf87374feca5cd-395d9a8ab2d0be3"
          },
          {
            "name": "14", "id": "c66ce186e85e8afc-97b3af05834f2899"
          },
          {
            "name": "15", "id": "3207c8beb2879bd-8027232ed069561cf"
          }
        ]
      }
    }
  ]
}
```

```

8027232ed069561cf},
  {"name": "16", "id": "236383ab351e476fcddec0d53cc3d2c10"}
},
{
  "id": "24c884b29af05007a81b4fb835fe79bf",
  "type": "Number",
  "slug": "current-payment-stage",
  "displayName": "Current Payment Stage",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Current Payment Stage",
  "validations": {
    "format": "integer",
    "precision": 1,
    "allowNegative": false,
    "minValue": 1
  }
},
{
  "id": "751887ce5476d458c9c1812706ac184d",
  "type": "PlainText",
  "slug": "payment-amount-for-current-stage",
  "displayName": "Payment Amount for current stage",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Payment Amount for current stage",
  "validations": {
    "singleLine": true
  }
},
{
  "id": "6d987a89973bd8069f0ff144ba83d581",
  "type": "PlainText",
  "slug": "invoice-number",
  "displayName": "Invoice Number",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Invoice Number",
  "validations": {
    "singleLine": true
  }
},
{
  "id": "7a93b0c837ffed52cdd882c221777311",
  "type": "DateTime",
  "slug": "invoice-generate-date",
  "displayName": "Invoice Generate Date",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Invoice Generate Date"
},
{
  "id": "df818683bfe29640dfcd8b3fd58f1e11",
  "type": "DateTime",
  "slug": "invoice-validation-date",
  "displayName": "Invoice Validation Date",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Invoice Validation Date"
},
}

```

```

{
  "id": "de210becf9a0307f2747a156ece1c834",
  "type": "File",
  "slug": "attachments",
  "displayName": "Attachments",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Attachments"
},
{
  "id": "0ff5db73cc8d8df75f9ea108b7869351",
  "type": "RichText",
  "slug": "client-notes",
  "displayName": "Client Notes",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Client Notes"
},
{
  "id": "f379a028e5642994fbae4d3676e3ca0b",
  "type": "Option",
  "slug": "select-language",
  "displayName": "Select Language",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Select Language",
  "validations": {
    "options": [
      {"name": "English", "id": "9e8d48cd0a6c9d24402e3c67ea9090f2"},
      {"name": "German", "id": "90942bf7bd1805ec704bb093dbacf2d0"},
      {"name": "Serbian", "id": "d491a2210036bb2ac41d58bcf6affe86"}
    ]
  }
},
{
  "id": "133c2fe2b393752f3141c63493565555",
  "type": "Switch",
  "slug": "client-active",
  "displayName": "Client Active",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Client Active"
},
{
  "id": "eaceeabe18756ca3d1a88097c4a4e4a1",
  "type": "Switch",
  "slug": "invoice-received",
  "displayName": "Invoice Received",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Invoice Received"
},
{
  "id": "8d6876e650063254127a475883c2f2ba",
  "type": "Switch",
  "slug": "payment-received",
  "displayName": "Payment Received",
  "isRequired": false,
  "isEditable": true,
  "helpText": "Payment Received"
},
{
  "id": "96b3cdbe880400091f9e304292713408",

```

```

    "type": "Switch",
    "slug": "follow-up",
    "displayName": "Follow Up?",
    "isRequired": false,
    "isEditable": true,
    "helpText": "Follow Up?"
  },
  {
    "id": "34de635456f1a54e55da1bfefd5fe2e0",
    "type": "Switch",
    "slug": "upcoming-job",
    "displayName": "Upcoming Job",
    "isRequired": false,
    "isEditable": true,
    "helpText": "Upcoming Job"
  },
  {
    "id": "6d378fc85585eeee66773d3df7c298cd",
    "type": "Switch",
    "slug": "currently-being-worked-on",
    "displayName": "Currently Being Worked On",
    "isRequired": false,
    "isEditable": true,
    "helpText": "Currently Being Worked On"
  },
  {
    "id": "31c3da56df5332fc8445038aee2cc2f5",
    "type": "Switch",
    "slug": "job-completed",
    "displayName": "Job Completed?",
    "isRequired": false,
    "isEditable": true,
    "helpText": "Job Completed?"
  }
]
}

```