

Product Manager Intern Assignment

As a Product Manager at Cloudflare, you'll partner closely with engineering to rapidly prototype ideas and build a deep understanding of the user experience.

For this exercise, you'll act as both **product thinker** and **hands-on builder** to:

1. Part 1: Build a quick prototype of the Build Challenge.
2. Part 2: Provide actionable, critical feedback on the products you use.

The outcome of your **prototype is secondary to the insights you gather** while building it. We value a broken prototype with a brilliant product critique over a perfect prototype with no feedback.

Part 1: The Build Challenge

As a product manager, your team gets product feedback flowing in from many places every day: Customer Support Tickets, Discord, GitHub issues, email, X/Twitter, community forums, and more. It's noisy and scattered, making it difficult to extract themes, urgency, value, and sentiment.

Your assignment is to **prototype a tool that helps aggregate and analyze feedback** so you can derive meaningful results from the feedback that you receive.

There is no single right approach for what the solution could look like: it could be a dashboard that shows the aggregated feedback, an AI agent that reports back the results, a workflow that sends results to your Slack or Discord with daily feedback, or something more unconventional.

Build Requirements

Vibe-coding tools (e.g., Claude Code, Windsurf) help PMs quickly bring ideas to life and are used regularly by PMs at Cloudflare. We **highly encourage** you to use these platforms to help build your solution.

With your solution, you are expected to:

- Host and deploy the solution on [Cloudflare Workers](#) (see tips below!)
- Using 2-3 other [Cloudflare Developer Platform](#) products is *strongly preferred*
- Using mock data is fine, no real integration with third party platforms is required
- Provide a short architecture overview that explains which Cloudflare products you used and why

Part 2: Cloudflare Product Insights

Throughout this exercise, you should use Cloudflare products to build out your prototype. You will experience what it's like to be a customer who uses Cloudflare first hand. As you build your prototype, we want you to maintain a **"Friction Log" of your experience using Cloudflare to build your project**. You can consider the onboarding and set up experience, documentation, and UI/UX as a few examples.

Response format

Please provide at least **3-5** insights using the format below.

- **Title:** [Concise name of the issue]
- **Problem:** [Describe what happened or what you noticed. Was it a technical bug, a confusing UI element, or a gap in the documentation? How did it slow you down?]
- **Suggestion:** [As a PM, how would you fix this? Is it a UI change, a new documentation section, a better error message, or a completely new feature?]

Expected Effort

We are not expecting a perfect prototype, and the focus of the project is to see how you think through the problem and solution.

The expected effort is 3-4 hours. We recommend the following time-boxing:

- 1 Hour: Familiarize yourself with the Cloudflare Developer Platform and docs
- 2 Hours: Build the prototype (we strongly recommend to not spend more than 2 hrs on the prototype)
- 1 Hour: Synthesize and write your Product Insights

Deliverables & Submission Instructions

For the assignment, please include the following information in a **pdf file**:

- **Project Links:** A link to a prototype demo that is deployed to Cloudflare Workers (e.g. your-project.account.workers.dev) and the GitHub repository of your prototype
- **Cloudflare Product Insights:** Your list of friction points and suggestions for how you would improve the Cloudflare products you used.
- **Architecture:** Short architecture overview that explains which Cloudflare products you used and why (hint: Take a screenshot of your Workers Binding page on the dashboard!)
- **Vibe-coding Context (Optional):** Short description of which vibe coding platform you used and some of the prompts that you used to build the solution.

To submit, please return to the original “**Product Manager Intern (Summer 2026) Assignment**” email and click the **Submit Here** link at the bottom. Attach your completed PDF where it states **Attach**, then click **Submit**.

Submissions will be reviewed on a rolling basis; the sooner you submit, the earlier we will get back to you. However, please ensure your project is submitted no later than **11:59 PM GMT on February 1, 2026**.

These instructions should have all the information needed to complete the assignment independently. If you experience any operational or submission issues, please reach out to us at pminternassignment@cloudflare.com.

Guide + Tips

If you are a student in the US with a valid .edu email address, you can receive one year of Cloudflare Workers upfront cost waived. See our [Cloudflare for Students](#) program.

Getting Started on Cloudflare Workers

The fastest way to start building is via the [CLI](#). In your terminal, run: `npm create cloudflare@latest`

We recommend running your coding agent (e.g. `claude`, `opencode`) inside this project to get started.

When you are ready to see your project live, run: `npx wrangler deploy`

Other Developer Platform Products

You can connect your Worker to other developer platform products via [Bindings](#) (how your Worker talks to other products) in your `wrangler.jsonc` or `wrangler.toml` file. Here are the products most relevant to this challenge:

- [Workers AI](#): Run machine learning models (like Llama 3) to summarize long feedback threads or perform sentiment analysis.
- [Workflows](#): Create multi-step, stateful workflows. Perfect for orchestrating a feedback pipeline (e.g., Receive Feedback → Analyze with AI → Send to Slack).
- [D1 Database](#): Cloudflare’s native serverless SQL database. Use this to store and query structured feedback entries.

- [AI Search](#): A fully managed retrieval augmented generation (RAG) pipeline that lets you perform semantic search which you can use to find "similar" customer complaints or themes.
- [KV \(Key-Value Storage\)](#): A highly-performant, eventually consistent key-value store.
- [R2 \(Object Storage\)](#): Store unstructured data, such as images or pdfs.

Pro Tip for Vibe-Coding: If you are using a tool like Cursor, Windsurf, or Claude Code, you can connect to our [Cloudflare Docs MCP](#) server to have the coding agent retrieve from the developer documentation.