

Setting Up Cursor for Trial Development

Author: Prof. Dan Pacheco, drpachec@syr.edu

Date: July 23, 2026. Delivered at NAHJ 2026 Conference Session.

Follow my latest updates at <https://danpacheco.com>

Part 1: Installing Cursor

1. Go to [Cursor.com](https://cursor.com)

2. Download the Cursor app (Mac or Windows)

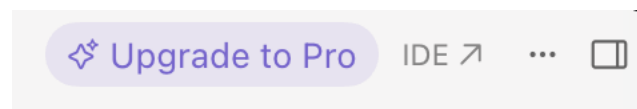
3. Click Sign Up and create an account.

- Skip the Pro Plan for now.

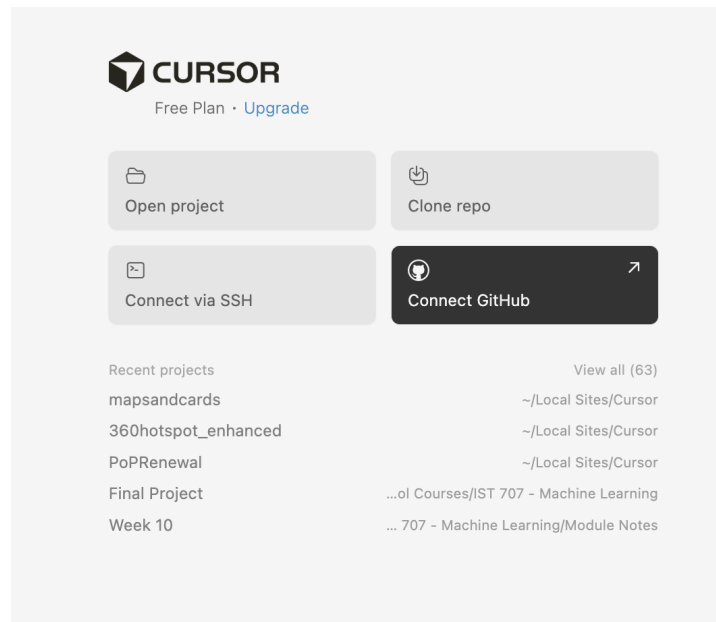
- Also skip the Github integration for now. We're only developing locally.

4. Trickiest part: Switch to the IDE, which means “Integrated Development Environment.”

- At the upper right of the app, you will see this. Click IDE.



5. You should now see this. Click Open Project, create a folder anywhere on your computer, and select it.



Congrats! You are now set up to develop on Cursor under a trial plan.

Part 2: Develop using an existing Github repository, like Maps and Cards.

This is actually where it gets easy. Just go into the chat window at upper right and ask it this:

PROMPT: “I want to develop a custom app using this Github repository:
<https://github.com/pachecod/mapsandcards>

Can you set that up for me to develop locally?”

It will download it and set it up in the folder you created. In a short time, it will give you a link to a Vite server. That's a local web server it has set up for you to test this app fully on your computer. Click it!

It will look something like this: <http://localhost:####> with #### being a four-digit port number.

Tell it to run it locally.

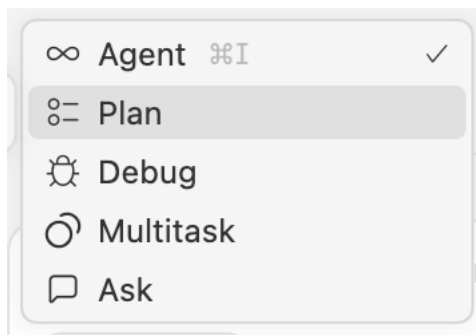
PROMPT: "Let's try running it locally."

You will then see something like this in a browser. The full app should be running on your local computer. Test it out.

Now, you're ready to add some customizations. Try this:

PROMPT: "Please describe the annotation types that are possible and suggest some new ones that would be useful for a tool like this."

Ask it which of the features it suggested will be easiest to deploy and test. Alternatively, you can give it your own ideas. In that case, you want to switch from Agent mode to Plan Mode:



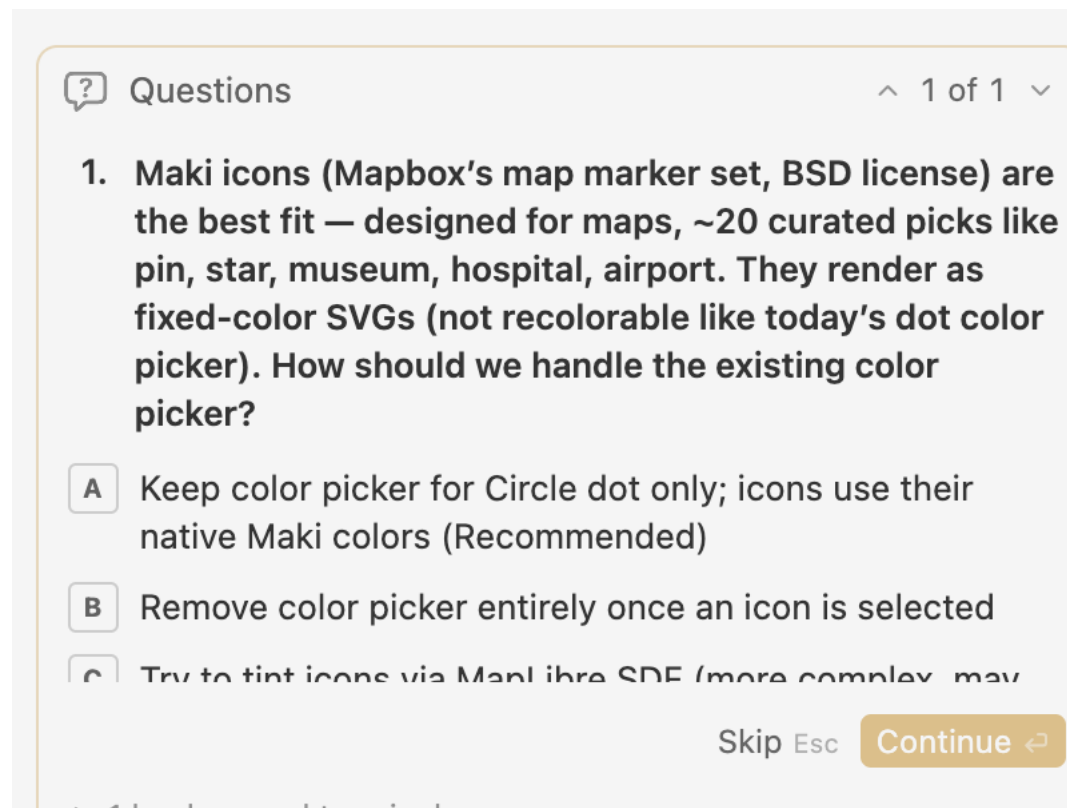
Here is an example of that:

PROMPT: "I like the idea for Icon Markers. I would like about 20 such markers, and they need to come from an open source / free licensed provider of such icons. The user should

be able to choose and change any marker from that collection using a pulldown that shows multiple markers in a tile form. Make a plan, please!”

Notice how detailed the prompt is above. These are basically what in software development are referred to as Product Requirements, or Product Specifications. You need to be extremely clear about your vision. You can even include sketches and drawings of what you want, literally from the “back of a napkin” or from whatever you like to use to create wireframes. Even boxes and lines from Powerpoint will work.

It may come back with questions, like this. If you like the answers, pick one. If not you can cancel that or click Other and give it more direction.



The image shows a 'Questions' dialog box with a title bar containing a question mark icon and the text 'Questions'. To the right of the title bar is a close button (X) and a dropdown menu showing '1 of 1'. The main content area contains a numbered list of three options:

- 1. Maki icons (Mapbox’s map marker set, BSD license) are the best fit — designed for maps, ~20 curated picks like pin, star, museum, hospital, airport. They render as fixed-color SVGs (not recolorable like today’s dot color picker). How should we handle the existing color picker?**
- ☐ **A** Keep color picker for Circle dot only; icons use their native Maki colors (Recommended)
- ☐ **B** Remove color picker entirely once an icon is selected
- ☐ **C** Try to tint icons via MapLibre SDF (more complex, may

At the bottom right of the dialog box are two buttons: 'Skip Esc' and 'Continue' with a right arrow icon.

When its plan is done, you will see something like this. Click “View Plan” to see what it had in mind:

Created Plan

Icon Marker Picker

Add ~20 BSD-licensed Maki map icons as an alternative to the default colored circle marker, with a tile-grid dropdown picker in the builder and MapLibre symbol-layer rendering in both builder and story viewer. Circle dot keeps the existing color picker; icons use native Maki colors.

View Plan

Build ⌘ ↩ ✓

In the center you will see something like this. Review it and make sure it matches your vision. If not, refine it in the chat window.

Icon Markers with Tile Picker

Icon source

Use **Maki** (BSD 2-Clause) — Mapbox’s open map-marker icon set, purpose-built for cartography.

- Vend 20 SVG files into `Tools/icons/maki/` plus `Tools/icons/maki/LICENSE.txt` (BSD attribution).
- Curated set (map-story friendly):

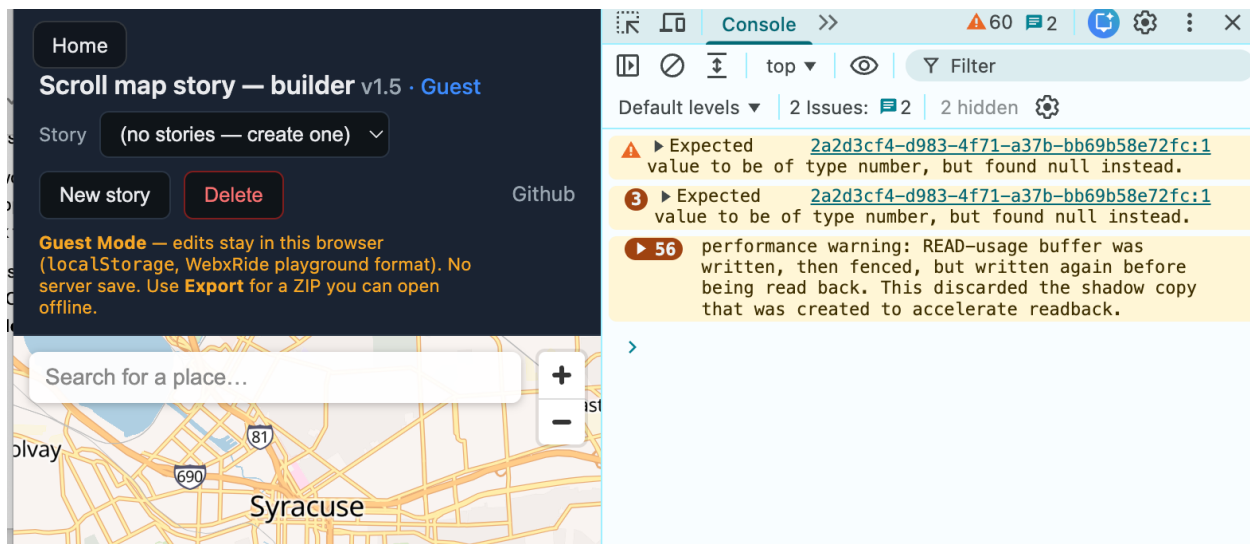
ID	Label
<code>circle</code>	Circle dot (default; not a Maki file — rendered by existing circle layer)
<code>marker</code>	Pin

When you like the plan, click Build and watch the magic! It should give you a link to view it in Vite server. If not, ask it:

PROMPT: “Can you spin this up in a Vite server for me?”

Try out what is created. It won’t be perfect the first time. If something isn’t working, tell it: When I do this, nothing happens, or this happens but this other thing should happen. Just talk to it like you would talk to a developer doing work for you.

One thing that may help it is if you share what are known as “Console Errors” in the test browser you are using. On Chrome, you do this by clicking **Command-Option-i** on a Mac, or **Control-Shift-I on Windows**.



You will want to right-click the errors and copy, and just paste that into cursor. It will find what it needs.

At this point you may hit your usage limit. Upgrade to Pro and pay for one month, you won't regret it and you can cancel it for that month later if you want.

Part 3: Publishing What You Made

Once you have something you really like, you need to get it on a server. The one I suggest using is Render.com because it works really well, and it also has a free tier.

1. Go to Render.com and create a free account.
2. With Render logged in, go back to Cursor and just tell it something like this:

PROMPT: "I want to deploy this to Render. Walk me through how to do that."

It may just set things up so it can log in through you. A safer way is to first ask it to publish to a Github repository that is private, and then have Render pull from that. That way, you can push to the github and have history of changes and roll back when needed.

Congratulations, you are now an AI-powered product developer!