Clone any app you pay for

Here's how to set up what I showed in my video. You install Replica, a free set of Claude skills, then point it at an app you pay for. Claude writes the code.

What you need

- A paid Claude plan. Claude Code comes with Pro, Max and Ultra (and with Team and Enterprise). It isn't in the free plan.
- The Claude desktop app. The skills run in **Claude Code**, which is the **Code** tab of the app, not the regular chat.
- A computer. The skills' small tools need Python 3.8 or newer. On a Mac, if a window asks you to install the command line developer tools, click Install: it's Apple's free package, and it includes Python.

- To put your app live: a domain of your own, from any domain registrar, and a hosting account (Vercel, unless you pick another; its free plan is for personal, non-commercial use). You buy the domain and do the sign-ins yourself. Claude never buys anything or types your passwords. It tells you exactly what to set.

Using Codex instead? The steps are at the end of this guide.

What it does

Replica is eleven free skills for Claude, made by Jake Schincariol: https://github.com/Jakeschincariol/replica-skill. It's open source under the MIT licence, with no signup and no API key. A skill is a saved set of instructions Claude follows when you call it.

Think of it as a mini software factory. Each skill does one job and leaves its work in a folder called **replica** in your project, where the next skill picks it up. Together they map the app, plan it, rebuild its look, build it, add sign-up and payments, test it, score it against the original, find what real users hate about

it, give it a brand of its own, write the landing page and put it live. The steps below go through them in order.

It rebuilds what an app does, never what it owns: the features and the flow, not its code, logo, words or content. You can rebuild a music app's player and playlists, not its catalogue. And it won't promise a perfect copy. A small tool can take a weekend, and a booking tool takes weeks. The first skill tells you honestly which one you're looking at.

Step 1: open Claude Code

1. Download the Claude desktop app from claude.ai/download, install it and sign in.
2. Click **Code** at the top of the sidebar.
3. Make a new folder for your clone, for example "My clone" in your Documents.
4. In the row under the message box, choose **Local**, then click the folder control, choose **Open folder...** and pick that folder.
5. In the same row, choose **Auto**, so Claude can keep going without stopping to ask at every step.

A full build has a lot of them.

> Keep an eye on it. In Auto, Claude can install programs and edit or change files without asking first. If you'd rather approve each step, choose **Manual**.

Step 2: install the skills

Paste this into the message box and press Enter:

```
Install the Replica skills from
https://github.com/Jakeschincariol/
replica-skill for me as personal skills, so I
can use them in any folder.
```

Claude adds the eleven skills for you. When it's done, start a new session in the Code tab so they load, then type `/replica` to see them in the list.

Rather do it yourself? The skills' GitHub page shows the other ways to install them, under "Install".

Step 3: pick an app and map it

Pick an app you pay for, and start small: the one thing you actually pay it for. “All of Notion” is too big. “Notion’s pages and sharing” is a project. Then type `/replica-recon`, the app’s website or app store link, and what you want. For example:

```
/replica-recon [the app's link] I want to
build my own version of its [the one thing you
use it for], just for me.
```

It asks which app and platform, which part of it, and who it’s for. Then it reads the app’s public pages, and you can click through your own account with it. It never logs into anyone else’s account and never gets past a paywall. If your account’s terms forbid building a competitor, it says so and sticks to public pages. You get a map of every screen and flow, and a size: a weekend, a few weeks, or a quarter.

Step 4: let it build

Run these one after the other, each when the last one is done:

1. This plans it:

   ```
   /replica-architect
   ```

2. This rebuilds the look:

   ```
   /replica-design
   ```

3. This builds it, screen by screen:

   ```
   /replica-build
   ```

4. This adds sign-up, the database and payments. If your app takes payments or connects to another service, you create those accounts yourself. Claude tells you what to set up.

   ```
   /replica-backend
   ```

5 This tests every flow for bugs:

```
/replica-test
```

6 This scores it against the original and lists what's left:

```
/replica-diff
```

Step 5: make it better than the original

Run this next:

```
/replica-entrepreneur
```

It reads real reviews of the original, from the app stores, review sites and Reddit, and ranks what users hate, what they keep asking for, and who the app ignores. Every quote is real and linked. It turns the top problems into a fix plan for your clone, and an angle you could sell it on. Then ask Claude to build the fixes.

Not sure which app to clone? Run this one first, on its own. It tells you whether there's a gap worth filling before you build anything.

Step 6: make it yours and put it live

1. This names it, with the trademark and domain checks to run, gives it new colours and a voice, and checks that nothing of the original is left in your code:

   ```
   /replica-brand
   ```

2. This writes the landing page, the pricing and the app store listing:

   ```
   /replica-launch
   ```

3. This checks everything first: the tests, what's missing, the rebrand. Then it gives you the settings for your domain and ships it when you say go.

   ```
   /replica-deploy
   ```

Step 7: add the features you always wanted

Now it’s your app. Ask Claude for whatever the original never gave you. For example:

```
Add [the feature you've always wanted].
Then run /replica-test again so nothing
else breaks.
```

Before you sell it

> Run the trademark checks `/replica-brand` lists, read the terms of anything you used, and talk to a lawyer before launch if there’s money on the line. This guide isn’t legal advice.

Using Codex instead of Claude Code

Replica is written for Claude, and its GitHub page only covers Claude. All eleven skills load in Codex, but I haven’t built an app with them there, so treat

this as a first try. You'll need a ChatGPT account and the ChatGPT desktop app.

1. Download the ChatGPT desktop app from chatgpt.com/download, install it and sign in.
2. Choose **Codex** from the ChatGPT dropdown, then open a new folder for your clone.
3. In the permissions control below the message box, choose **Approve for me**, so it can keep going without stopping to ask. Keep an eye on it, because it can install programs and change files.
4. Paste this into the message box and press Enter:

   ```
   Install the Replica skills from
   https://github.com/Jakeschincariol/
   replica-skill for me as personal skills
   in ~/.agents/skills, so I can use them in
   any folder.
   ```

5. Start a new chat so the skills load.

Then follow Steps 3 to 7. Instead of typing a command like `/replica-recon`, ask for the skill by name. For example:

> `Use the replica-recon skill on [the app's link]. I want to build my own version of its [the one thing you use it for], just for me.`

Written on 7 November 2026 from version 1.0 of Replica. Claude changes fast: if a menu has moved, look for the same words in the app.