

AGENT BUILDERS

Build with Codex

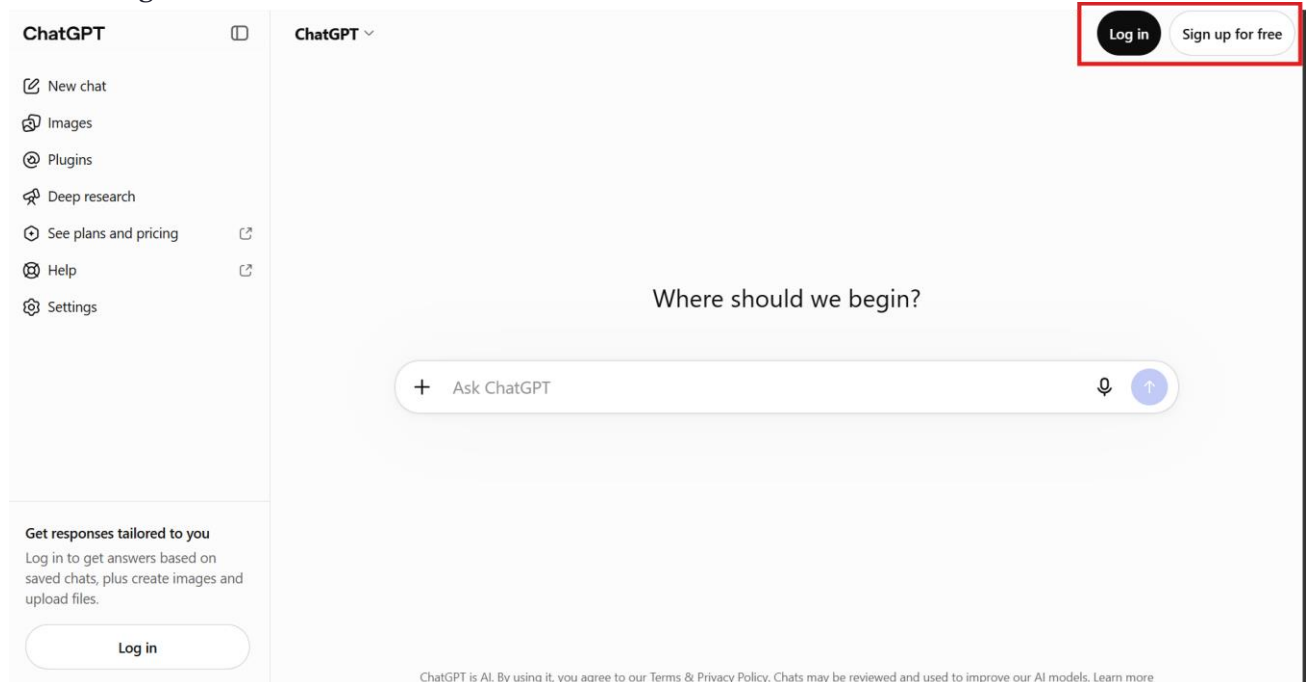
A practical guide to building, changing and testing projects with Codex.

Agent Builders · Codex workshop

If you do not have ChatGPT/Codex yet

You need a ChatGPT account and the ChatGPT desktop app. A free account is fine.

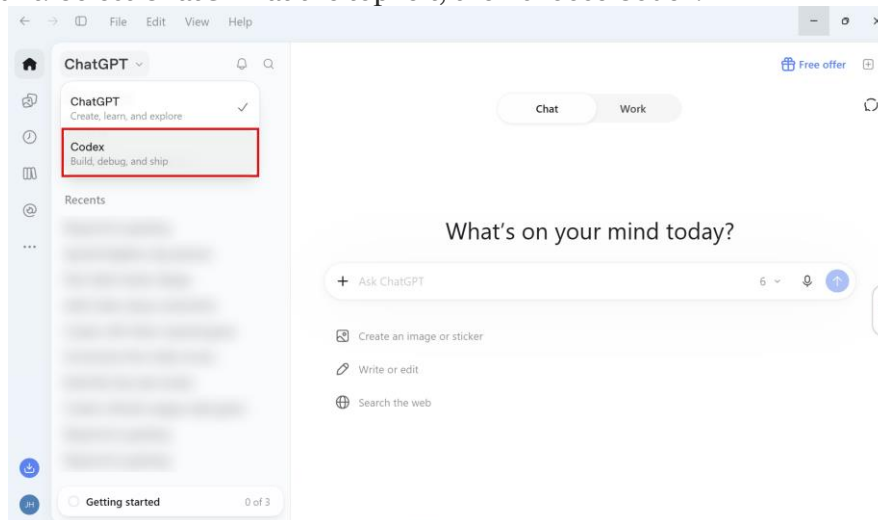
1. Go to [ChatGPT](#) and select **Sign up for free**. If you already have an account, select Log in instead.



The sign-up and log-in buttons are in the top-right corner.

2. Create your account and sign in. Use an email address you can access during the workshop.
3. Download and install the ChatGPT desktop app for Windows or macOS: [Download the ChatGPT desktop app](#).

-
4. Once it has finished downloading, open the desktop app and sign in with the same account. Select ChatGPT at the top left, then choose Codex.



Select ChatGPT at the top left, then choose Codex.

Before you start

What we are doing today

Over the next two hours you will build with Codex, each one using what you learned in the last.

- **Part 1 - Habit tracker:** plan it with Codex, build it, then add features of your own.
- **Part 2 - Make it a game:** add rewards to your tracker, such as coins and a shop.
- **Part 3 - Solve a problem:** start a new project from one of the challenges, or your own idea.
- **Showcase:** show everyone what you made.

The timings are a rough guide. Some people will get further than others and that is fine. Each part has extra ideas if you get through it quickly.

How to work

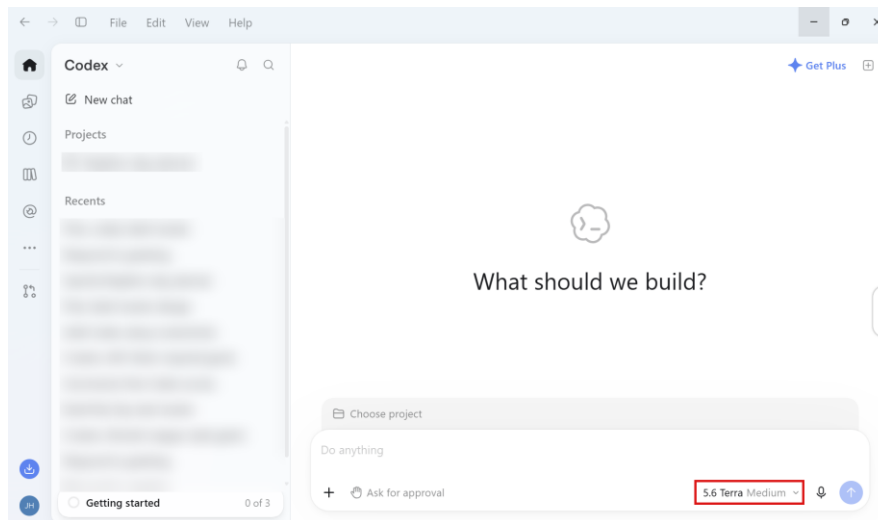
- Get a basic version working before adding extras.
- Make small changes and test each one.
- If something breaks, say exactly what stopped working.

Model and effort

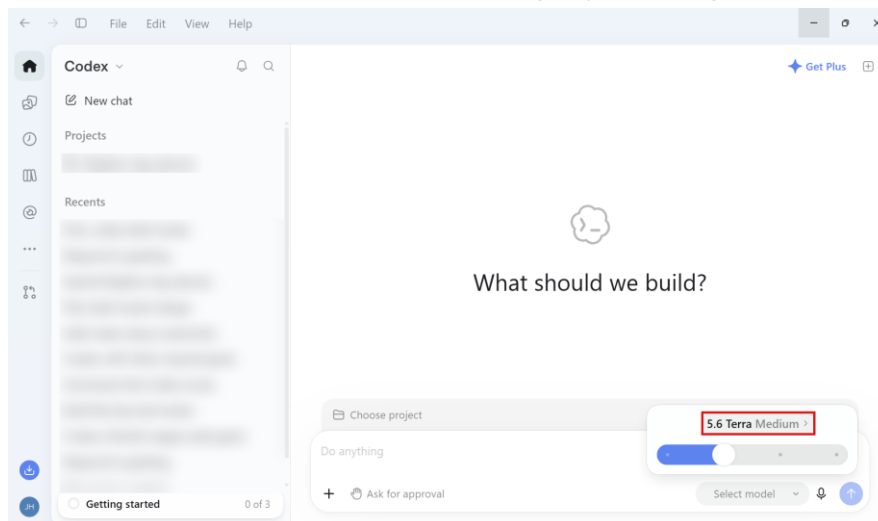
Before you start, change the model to 6 Luna and set the effort to Light. Codex usually starts on 5.6 Terra on Medium, which uses your allowance much faster, and once it runs out you

cannot keep building. If the results are not good enough, move the effort up to Medium. If your usage starts running low, move it back to Light.

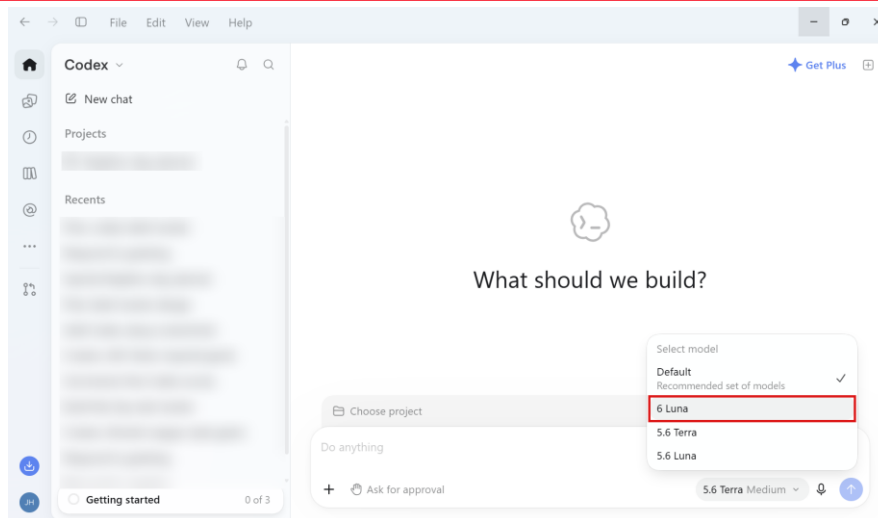
These steps and screenshots are for a free account. If you have a paid plan, see the note at the end of this section.



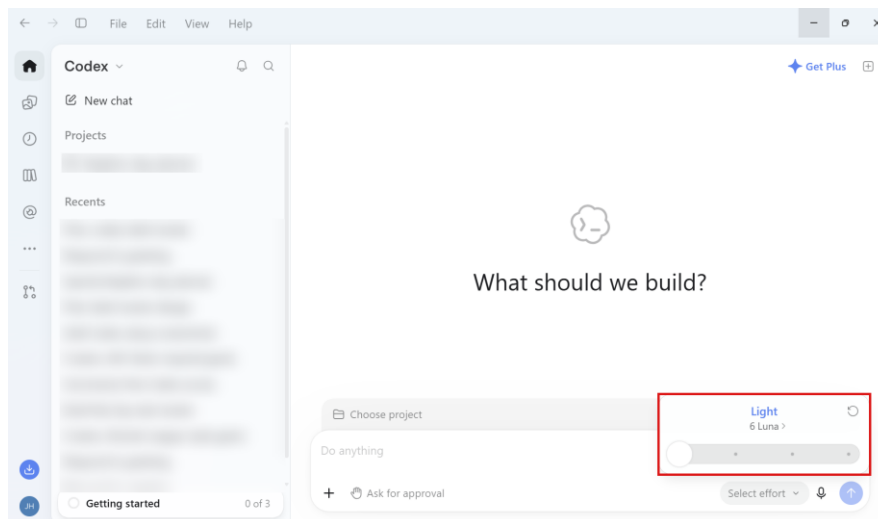
Select the model button at the bottom right of the message box.



Select the model name at the top of the panel to see the list of models.



Choose 6 Luna from the list.

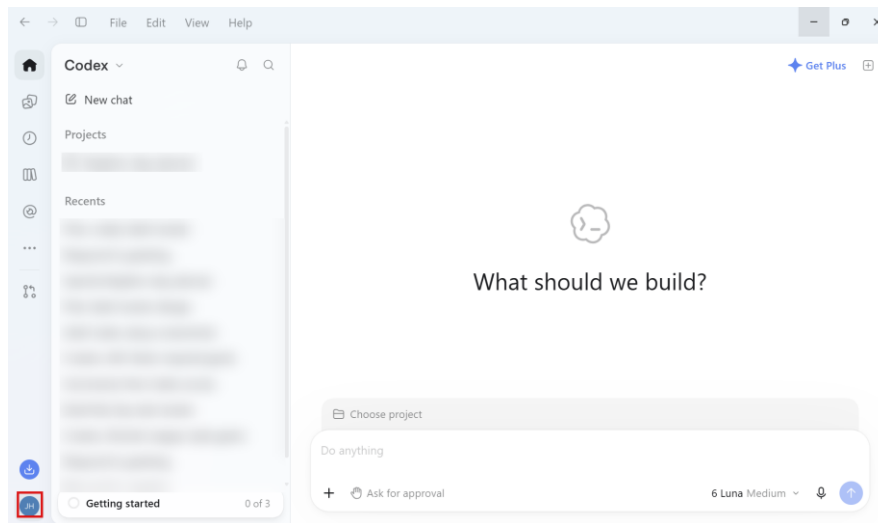


Move the effort slider all the way left to Light. Move it one step right for Medium.

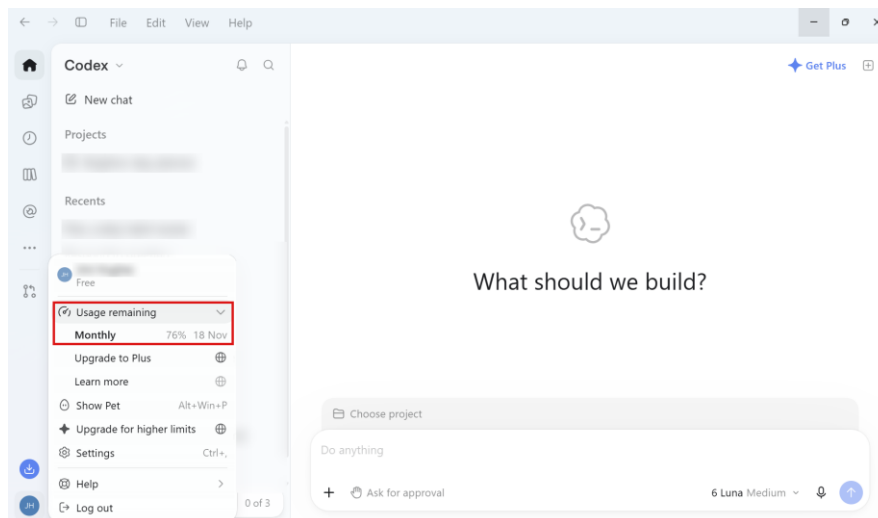
Paid account? Avoid GPT-6 Astra: You get more usage than a free account and more models to choose from, so your list may look different from the screenshots. 6 Luna on Medium is a good place to start, and you can try Sol if you get stuck on something harder. Avoid GPT-6 Astra as it uses your allowance much faster, and you could run out before the session ends. Whichever plan you are on, keep an eye on your usage using the steps below.

Check your usage

Keep an eye on how much you have left as you go. Select your initials at the bottom left of Codex, then look at Usage remaining. If it is dropping quickly, check you are on 6 Luna and move the effort back to Light.



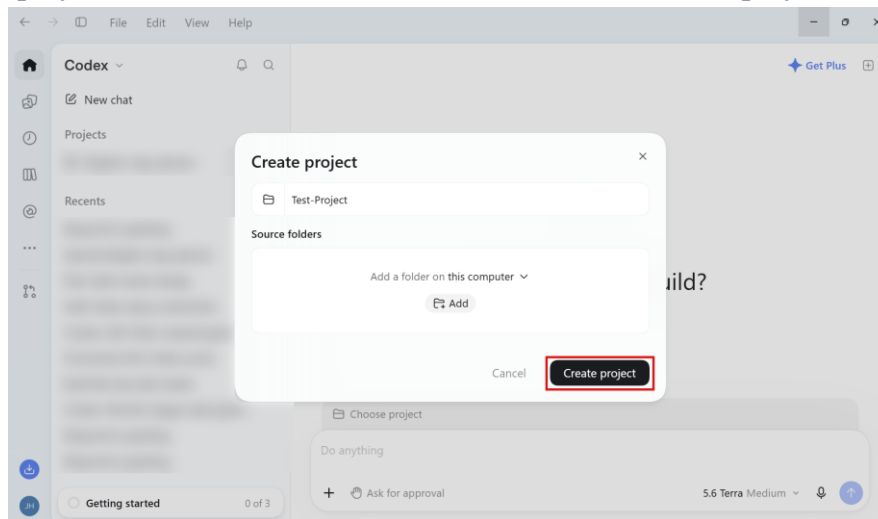
Select your initials at the bottom left.



Usage remaining shows how much you have left and when it resets.

Your first few minutes in Codex

Create a new project in Codex called habit-tracker, then select Create project.



Type a name for your project, then select Create project.

Check the model is on 6 Luna and the effort is on Light.

Find the message box. This is where you paste or type your prompts.

Codex may ask permission before it creates files or runs a command. Check it is working in your project, then allow it.

To see your app, ask Codex how to open it in your browser.

Plan before you build

A vague prompt is the quickest way to waste time and usage. Ask for "a habit tracker" and Codex has to guess the features, layout and colours. You then spend the next half hour asking it to undo things you never wanted.

A better prompt tells Codex:

- what you are building and who it is for;
- the main things it needs to do;
- how you want it to look;
- anything you do not want.

Then ask it to ask you questions and write a short plan before it writes any code. Read the plan properly. Changing it at this stage takes seconds. Changing it after it is built takes much longer.

Part 1 gives you a prompt that does this. By Part 3 you will write your own.

Part 1: Build a habit tracker *around 30 minutes*

Build an app for tracking daily habits. Everyone starts with the same prompt, but you decide the features and the look.

Step 1 - Plan it

COPY THIS:

```
I want to build a daily habit tracker to help me keep up good habits. I should be able to add habits, tick them off each day and see my progress. Before writing any code, ask me up to five questions about the features, layout, colours and anything I do not want. Then write a short plan for me to approve. Do not write code until I approve it. It should run in a browser using plain HTML, CSS and JavaScript, with no accounts or paid services.
```

Answer the questions with your own choices. If the plan includes something you do not want, tell Codex to change it before you approve.

Step 2 - Build the first version

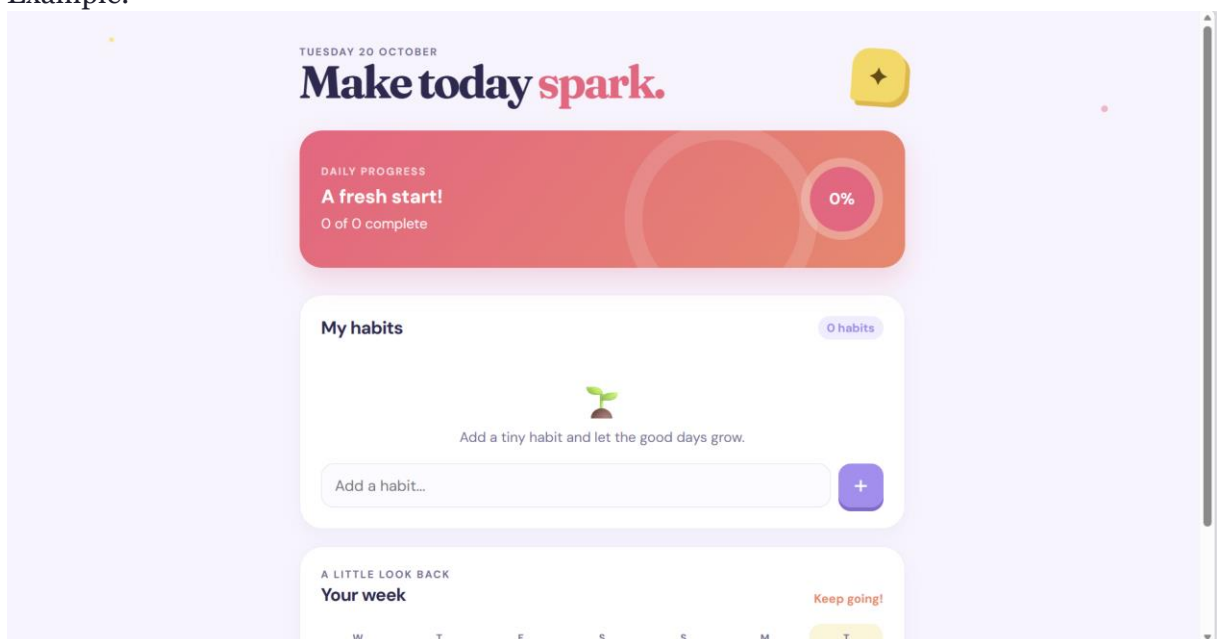
COPY THIS:

The plan looks good. Build the first working version. Make sure my habits are still there after I refresh the page. Tell me how to open it and what to test.

Step 3 - Check it works

Add a habit, tick it off, delete one and refresh the page. If anything is wrong, get it fixed before adding more.

Example:



Step 4 - Add two features of your own

This is the main part of the build. Pick two features from the list or come up with your own, then add them one at a time. Test each one before starting the next.

- streaks for habits done several days in a row;
- a weekly calendar view;
- a progress bar for today;
- habit categories or colours;
- editing habits;
- a message or animation when everything is done.

ADAPT THIS PROMPT:

Keep everything that already works. Add [feature]. It should [what it does and where it appears on the page]. Keep the current design and do not rebuild the app. Tell me how to test it.

If you have more time

Try statistics for the past week, habits that only happen on certain days, or a layout that works well on a phone.

Part 1 complete.

Part 2: Turn it into a game *around 30 minutes*

Stay in the same project and make completing habits more rewarding. A simple way is coins you earn for each habit and spend in a shop, but you can take it in any direction.

Step 1 - Plan the rewards

ADAPT THIS PROMPT:

I want to turn my habit tracker into a game. My idea is [your idea, for example coins for each habit I complete that I can spend in a rewards shop]. Look at the current app and suggest simple rules, including how to stop someone earning the same reward twice. Ask me any questions you need to, then wait for me to approve before changing any code.

Other ideas: a pet that grows, a garden, a room you decorate, a town you build up, badges or levels.

Step 2 - Add a way to earn

Build the earning part first. Complete a few habits and check your total goes up by the right amount. Untick and tick a habit again to make sure you cannot earn twice.

Step 3 - Add something to spend on

Add the shop, pet, garden or whatever you planned. If it would take days to earn enough to try it, ask Codex for a clearly labelled demo button that adds test coins.

Step 4 - Test it

Earn something, spend it and refresh. Check your progress and purchases are still there and the habit tracker still works.

If you have more time

Add achievements, levels, animations or a daily challenge. Or show your rewards as a scene that fills in as you unlock things.

Part 2 complete.

Part 3: Build something useful *around 35 minutes*

Create a new project in Codex, the same way you did for Part 1. This time you write the planning prompt yourself.

Step 1 - Pick a challenge

- Challenge A - Plan a day out:** Friends are visiting Brighton, or wherever you live. Build a planner that suggests activities based on their budget, interests and available time, adds up the cost and lets you change the plan.
- Challenge B - Fix an everyday annoyance:** Choose a small problem you regularly deal with, such as deciding what to cook, tracking spending, planning revision or splitting a bill. Build a tool that makes it quicker or easier.
- Challenge C - Make it for one real person:** Build something for someone specific in your life, such as a times-table game for a child, a big-button page for a grandparent or a

quiz for a friend. Think about the colours, wording and level of difficulty that would suit them.

- Challenge D - Build something you want:** Create a website, game or app that you would genuinely enjoy or use. Start with the simplest version you can finish and test today.

Not sure? Go with Challenge A. Already have your own idea? Use Challenge D. For Challenge A, ask Codex to create some example data and label it clearly as examples so you can start straight away.

Step 2 - Write your planning prompt

Use what you learned at the start. Say what you are building, who it is for, what it must do, how it should look and what you do not want. Then ask Codex to question you and write a plan. If you get stuck, start from this:

START WITH THIS:

```
I want to make a [website / app / game] that helps [who] with [problem]. It needs to [two or three main things]. I want it to look [style]. I do not want [anything to avoid]. Before you write any code, ask me up to five questions and write a short plan for me to approve. Use plain HTML, CSS and JavaScript that runs in a browser.
```

Step 3 - Build and test the main feature

Get the main feature working first. Try different inputs and check the results change the way they should. A planner that suggests the same thing whatever you enter is not finished yet.

Step 4 - Get someone else to try it

Let someone else use it for two minutes without help. Watch where they get stuck, then fix one thing. If you are online, describe what it does to someone in the chat and ask what they would add.

If you have more time

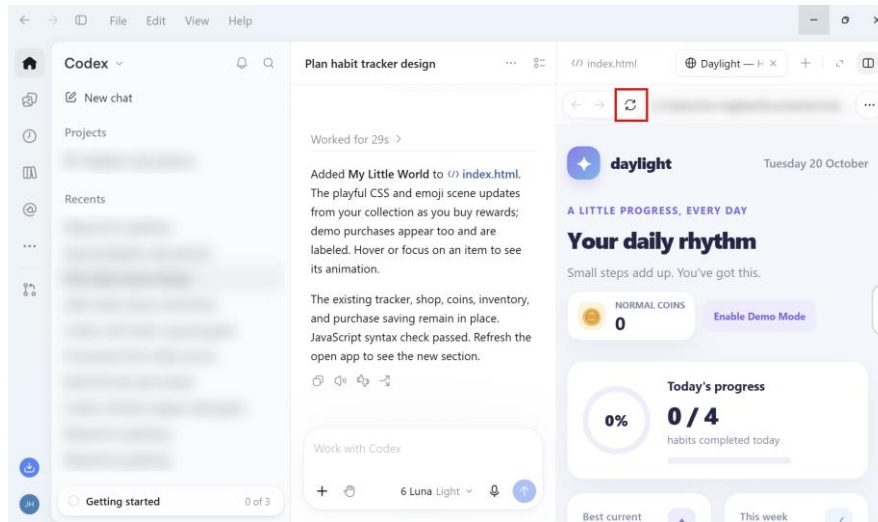
Load your own data from a CSV file, ask Codex to explain how the main logic works and then improve it, or make it work properly on a phone.

Part 3 complete.

If something goes wrong

Codex says it made a change but you cannot see it

Refresh the page. If Codex has opened your app in a panel inside Codex, use the refresh button at the top of that panel. If it opened in your browser, refresh the page there. If it still looks the same, ask Codex which file it changed.



The refresh button is at the top of the app panel.

Something has stopped working

Tell Codex what you clicked, what you expected and what happened. Paste any error message and ask it to fix that one problem. If its fix makes things worse, ask it to undo the last change and go back to the version that worked.

ADAPT THIS PROMPT:

I clicked [button] and expected [result], but [what happened instead]. Fix only this problem and do not change anything else that works. Tell me what you changed and how to test it.

Final test and showcase *around 15 minutes*

Stop adding things. Refresh, try the main features once more and make sure the text is easy to read.

When it is your turn, show us:

- what you built;
- one decision you made at the planning stage;
- a feature you added after the first version;
- one problem you found and fixed.

Joining online? Post a screenshot of your app in the meeting chat with a sentence about what it does.

Before you leave: save your work and keep the prompt that worked best.

Finished early? Keep going

Pick one of these rather than jumping between several.

- **Make it smarter:** have your app recommend something based on what the user enters.
- **Make it easier to use:** improve the phone layout, keyboard controls or navigation.
- **Try to break it:** test odd inputs, find three problems and fix one.
- **Build a game:** Breakout, a memory game, a quiz or a simple platformer.
- **Build a website:** a portfolio, a hobby page or a site for a local business.
- **Your own idea:** anything you have wanted to make.

ADAPT THIS PROMPT:

I want to build [idea]. Help me cut it down to a version I can finish and test today. Ask me up to five questions, then suggest a short plan with the most important features first. Do not write code until I approve it.