Stop making App Store screenshots in Figma

One HTML page per screenshot, one command for every iPhone and iPad size. The template is inside

THE SHORT ANSWER

Build each screenshot as a small web page: a feature name, a big headline, and your real app screen inside a phone frame.

A short script opens every page in Chrome without showing a window and saves it at **the exact pixel size Apple asks for.** iPad versions come out of the same pages. Change a headline, run it again, done.

Why Figma eats your evenings

Screenshots look like a one time job. They are not. Every time you add a feature, change a price or test a new headline, they need to change again. And the work multiplies fast:

WHAT YOU HAVE	IMAGES TO UPDATE
6 screenshots, iPhone only	6
Add iPad	12
Add 5 more languages	72
One headline changes	Every one of them, by hand

In Figma that means resizing frames, nudging text that no longer fits, and exporting again. In HTML it means editing one line of text.

How it works

shot_01.html
hook on top, your app screen below

→

render.sh
Chrome takes a picture of each page

→

iPhone + iPad
1206 × 2622
2064 × 2752

Same page, two sizes. The script just opens it at a different window size

This is what the template gives you on the first run, untouched. Left is the iPhone file, right is the iPad file. Both came from the same page.



A made up spending app, so you can see the layout. Swap in your own screen

WHAT YOU NEED

A Mac with Google Chrome. That's it. No paid tool, no design app, no plugin. If you can edit a text file you can do this.

Step 1. Grab the template

Download it here: **html-screenshots-template.zip**. Unzip it anywhere. You get three files:

FILE	WHAT IT DOES
base.css	Colors, the headline style, the phone frame and the iPad layout. Put your brand colors at the top
shot_01.html	One screenshot. Copy it to make shot_02, shot_03 and so on
render.sh	Turns every shot page into iPhone and iPad images

Step 2. Write the hook

The top of each page is the part people actually read while scrolling the store. Three lines, nothing more:

TOP OF SHOT_01.HTML

```
<div class="hook">
  <div class="eyebrow">Spending tracker</div>
  <h1>See where your<br><mark>money goes</mark></h1>
  <p class="sub">Every expense sorted the second you add it</p>
</div>
```

Whatever sits inside `<mark>` gets the tilted highlight box. Keep it to the two or three words that matter.

Step 3. Put your real screen in the phone

Below the hook is a phone frame. Inside it you rebuild your app screen with plain HTML: cards, numbers, a list. Use the same colors and font sizes as your app so it looks like the real thing.

Use believable data. "$1,284 spent" sells. "Lorem ipsum" and "Item 1, Item 2" tell people the app is empty.

Step 4. Get the iPad version for free

You don't make a second set of pages. Each page reads one word from its link, `#iphone` or `#ipad`, and the CSS switches to a bigger canvas and a tablet frame:

IN THE HEAD OF EVERY SHOT

```
<script>document.documentElement.className = location.hash.slice(1)</script>
```

IN BASE.CSS

```
html.ipad, html.ipad body { width: 2064px; height: 2752px; }
html.ipad h1 { font-size: 168px; }
html.ipad .phone { width: 1032px; height: 1376px; zoom: 1.62; }
```

Step 5. Render everything

RENDER.SH (THE CORE OF IT)

```
for page in shot_*.html; do
  for device in iphone ipad; do
    [[ $device == iphone ]] && size=1206,2622 || size=2064,2752
    "$CHROME" --headless=new --hide-scrollbars \
      --force-device-scale-factor=1 --window-size=$size \
      --screenshot="$PWD/out/$device/${page%.html}.png" \
      "file://$PWD/$page#$device"
  done
done
```

Open Terminal in the folder and run `./render.sh`. Your images land in `out/iphone` and `out/ipad`, and the script prints the size of each one so you can see it's right.

Keep `--force-device-scale-factor=1` in there. It locks every image to the exact size you set, whatever screen your Mac has. If a printed size is off by even one pixel, App Store Connect refuses the upload.

What Apple accepts

RULE	WHAT IT MEANS FOR YOU
iPhone 1206 × 2622	The 6.3 inch size. App Store Connect asks for at least one
iPad 2064 × 2752	Needed if your app runs on iPad, which iPhone apps usually do
Up to 10 per size	PNG or JPG
No transparency	Give the page a solid background and you're fine

Sizes change when new iPhones come out, so check Apple's screenshot specs page before a big release. If they change, you edit two numbers in the script.

Google Play: Play won't take a screenshot that's more than twice as tall as it is wide, so the iPhone size gets refused there. Add a third size to the loop, for example 1080 × 1920, with its own small CSS block like the iPad one.

Let AI rebuild your screens

Rebuilding a screen in HTML by hand is the slow part. I don't do it by hand. I give an AI coding tool my app's theme file and a screenshot of the screen, then ask:

> **PROMPT**
>
> ```
> Here is my app's theme file (colors, font sizes, corner radius)
> and a screenshot of one screen. Rebuild this screen as HTML inside
> the .app div of shot_01.html. Use only the colors and sizes from
> the theme file. Keep my real text and numbers. No placeholder
> data. Don't change base.css.
> ```

Then render, open the image, and fix whatever looks off. Two or three rounds is normal.

What to put on each screenshot

- **Shot 1 is your strongest promise.** Most people never swipe past it
- **Shot 2 names the problem** your user has, in their words
- **Then one feature per shot.** Never two
- **One thing that pops out of the phone** where it helps: a card, a big number, your mascot
- **Switch the background color** every shot or two so the row doesn't look flat in search

> ✎ **MY REAL SETUP**
>
> - All my newer apps get their screenshots this way now
> - On one of them it's 6 screens in 6 languages, so 72 iPhone and iPad images, plus the wide App Store banners, from the same pages
> - Translations sit in a small text file per language, and the pages read from it. I left that part out here, start with English
> - When a headline changes, I edit one line and run the script again. Nothing gets resized or exported by hand
> - Before uploading, I still open every single image. The script makes them, it doesn't check that text isn't cut off

Before you upload

- [ ] Brand colors set at the top of base.css
- [ ] Every shot has an eyebrow, a two line headline and one sub line
- [ ] Real screens with real looking numbers, no placeholder text
- [ ] Rendered with `--force-device-scale-factor=1`
- [ ] Sizes printed by the script match 1206 × 2622 and 2064 × 2752
- [ ] Opened every image and checked nothing is clipped or overlapping

Want more like this?

I build and ship apps next to a full time job, and I share what actually works, the boring tricks included. Over 100,000 people use my apps. Follow along wherever you hang out:

Instagram
3,500+ builders
@garoonoo →

LinkedIn
8,000+ followers
Connect →

Medium
250+ readers
Read the long versions →

See all my apps and free docs at garoono.in →

Go deeper

READ	Apple: screenshot specifications Every size App Store Connect accepts right now
READ	Chrome: headless mode What the Chrome flags in the script do
MORE DOCS	garoono.in/docs The rest of the free and Pro playbooks