

Y07 — How the Kit Works

By now you have built something real, called a real person, and put a real site on the internet. So you have earned the answer to a question some of you have been quietly asking: how does this actually work? Tonight we open the hood. Not to make you mechanics - to show you there is no magic in here, just a few simple parts. And people who know there is no magic stop being users and start being builders.

Part one: the website itself is just files. That whole beautiful page - the colors, the headlines, the footer - is a folder with a few text files in it. One file holds the words and structure. Another holds the styling - every color and size, written out like a recipe. When you told the AI to make the colors blue, it opened that recipe file and changed a few lines. That is all editing a website is: changing text in files. You could open them yourself and read them tonight.

Part two: the server, which is a fancy word for a waiter. Somewhere, a small program stands between your files and the world. A visitor's browser walks up and says, may I have the homepage? The waiter grabs that file from the folder and hands it over. That is the entire job. The one in your kit is about ninety lines long, and the reason your leader types `npm start` is literally just: waiter, clock in.

Part three: the safety check, which you already know as `npm run check`. Here is its secret: it is not smart. It is thorough. It walks through every file and asks dumb, careful questions: does every link point to a file that exists? Did anyone leave a bracket placeholder behind? Does the site actually start? Each question is simple; asking every one of them, every time, without getting bored - that is what computers are for. The check exists because the people who built this kit broke things in all those exact ways first. A checklist is just experience, written down.

Part four: the guardrails. Maybe you noticed the AI never touches certain files, and sometimes says, this needs your group leader. That is not the AI being cautious - it is a little instruction file in the project, telling it the house rules: only edit the website files, never the machinery, no personal info on pages, ask before the powered-by line moves. You have been working alongside written instructions this whole time. Whoever writes that file is the real boss of the project - which is worth remembering, because one day soon it could be you.

And part five, the cleverest trick in the kit: every place gets its own copy. The bakery's site and the church's site are not roommates; each one is a complete fork of the kit, separate and whole. That is why a Tuesday-night mistake on one site cannot touch any other, and why the church can someday hand the whole folder to the bakery and walk away clean. Independence is built in on purpose.

That is the whole machine. Files, a waiter, a checklist, house rules, and copies. Five parts, none of them magic. The kit did not make websites possible for you - it made failure boring. Every sharp edge someone could cut themselves on, the builders sanded down before you arrived. Which means the way you grow now is simple: start touching the edges again, on purpose, one at a time.

Your build for this lesson is a treasure hunt, and it is gentle. With your leader, open the project folder and just LOOK at three things: the file with the website's words in it - find a sentence you wrote through the AI, sitting right there in text. The waiter - scroll the server file end to end; you will not understand every line, and that is fine, just see that it is short. And the house rules file - read it top to bottom, because you now know enough to understand every single rule in it, and a month ago you did not. That is how far you have come.