

**How does a
website work???**

Firstly, here are our basic components of a website



1 Front-end ← 2 Back-end

3 Databases
/APIs

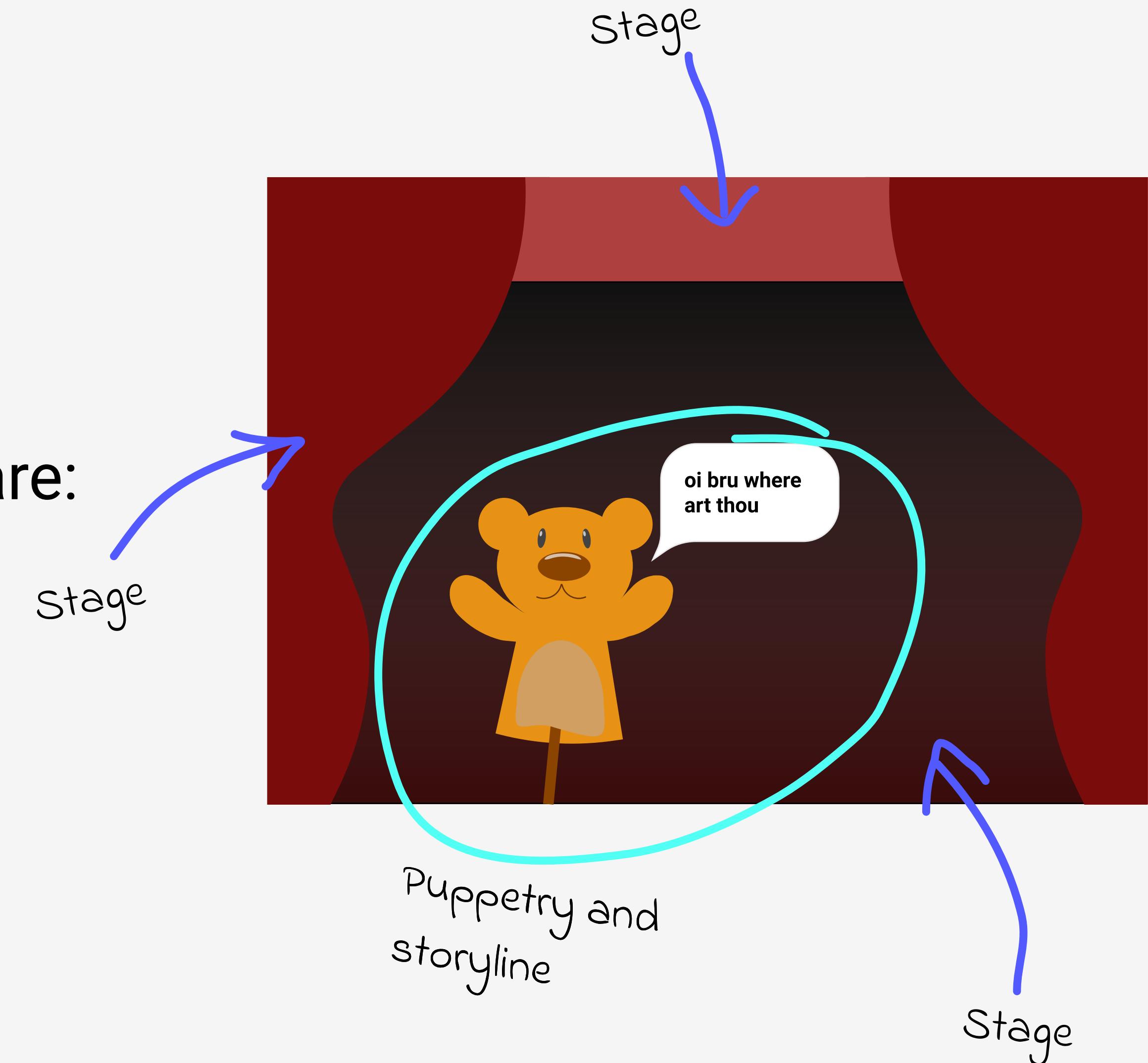
To better understand the relationship between the three, let's use an analogy.

To better understand the relationship between the three, let's use an analogy.

Imagine you are organising a puppet show.

The key components of the puppet show are:

- 1 • The stage
- 2 • The puppetry and storyline



The stage is what we call the 'front-end' in web development.

It's what you see 'in the front'.

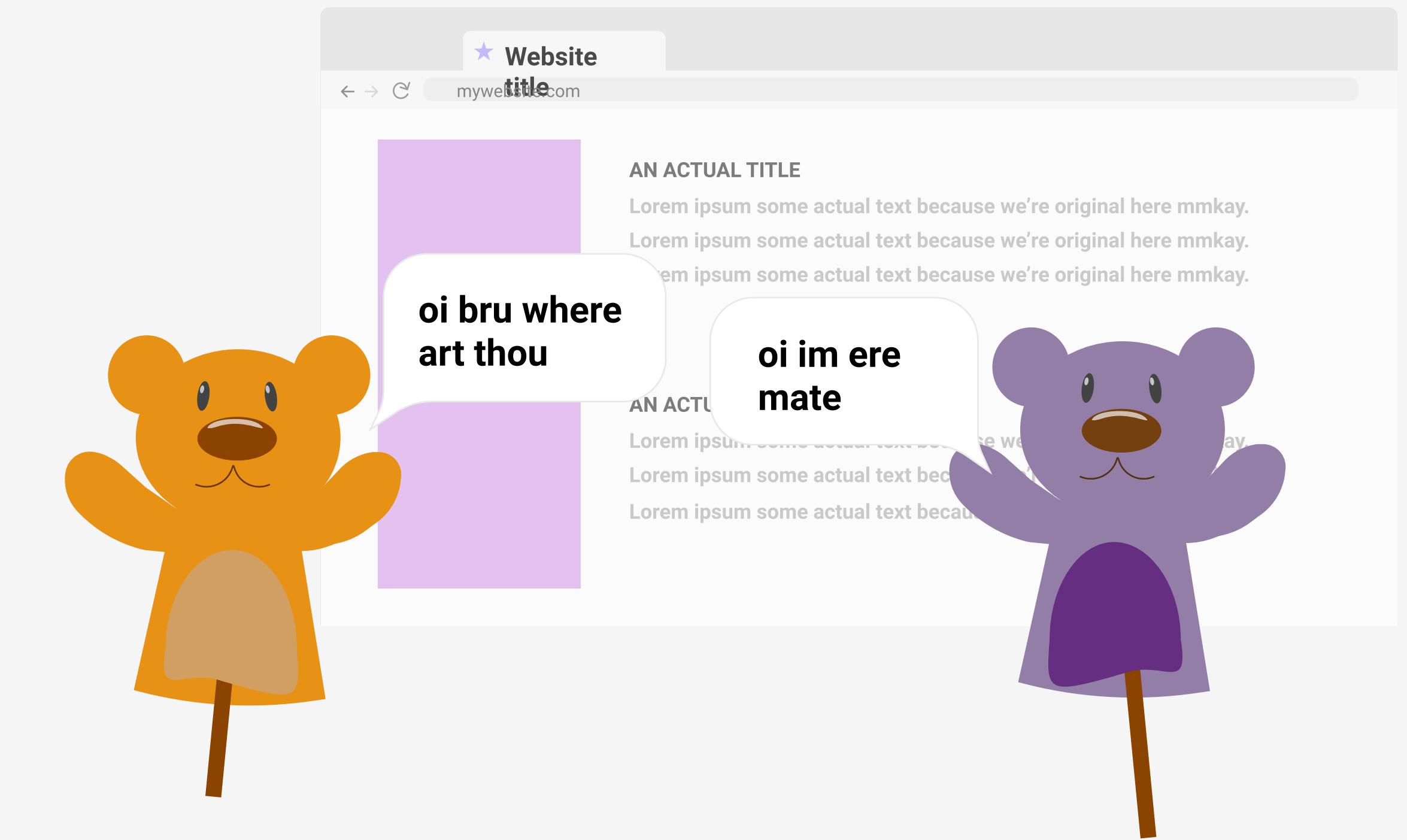
The visual interface.

- The visual objects, boxes, images
- Colours, actual text
- The structure of the page



The ‘back-end’ is what happens behind the scenes.

Things like the dialogues or person moving the puppets - what happens ‘behind the scenes’ - is the back-end.



Here are the key differences between the front-end and back-end.

Front-end

- Static
- Visual
- User interface

vs

Back-end

- Dynamically changes
- Information
- Actions

It's what you **interact** with

It **makes decisions** incl. what data to show you.

Why isn't puppet, a part of the stage since it's visual?

So hypothetically we could put a stuffed puppet in front of the stage.

But some things note about this - you wouldn't be able to change/move the puppet without disrupting the show.



^supposed to be a sticker..

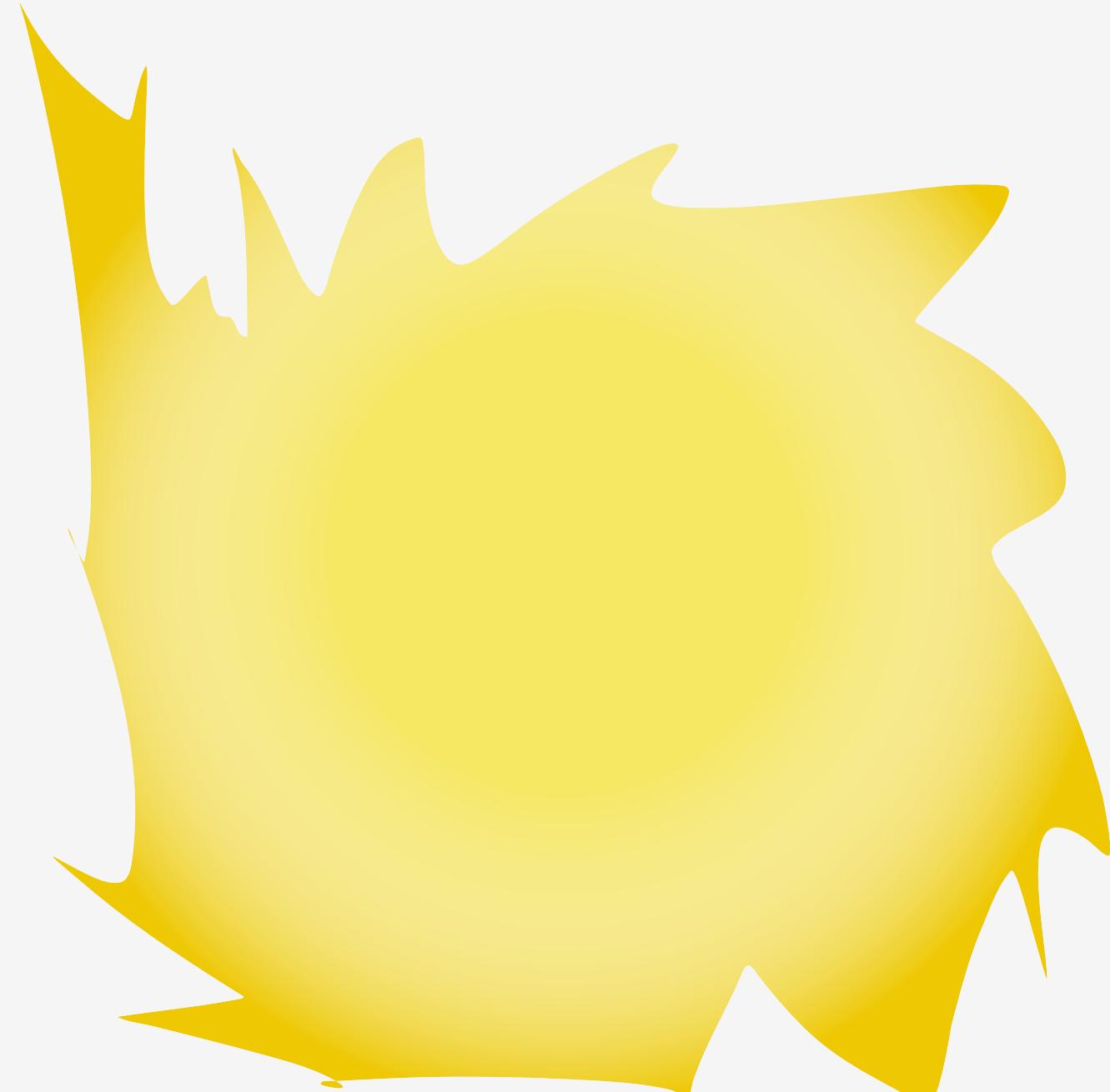
Dat is hard coding.

Adding a static puppet could suit some needs, for example perhaps you want the Teletubby sun in the corner?

But you are limited in the actions you want to take. The sun can't talk if it's just a stuffed toy with no actor behind it.

wtf

We call this 'hard-coding'. It's static data in the front-end.



Why not hard coding?

Let's say you wanted to do Shakespeare plays, but your puppet actors only know how to perform Potter Puppet Pals, specifically in a bogan accent.

Now, that wouldn't be very scalable would it?

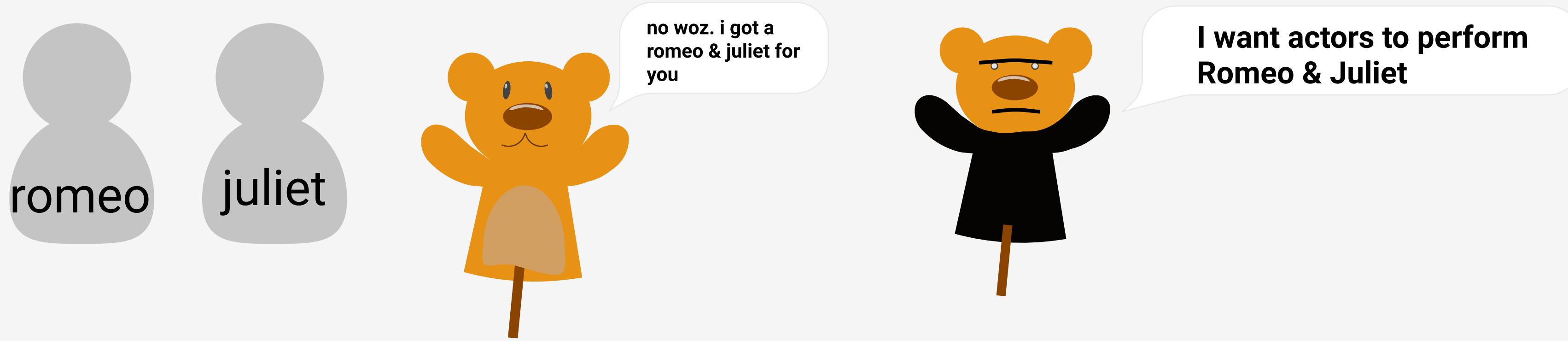
Your actors have a very specific set of skills for a specific purpose.

In the same way we want dynamically skilled actors, we also want dynamic data (not hard-coded data).



Tell me more about this d y n a m i c data

In order to show these Shakespeare plays, you outsource actors from a Puppet Actor Agency.



You are now able to pick the show you want, and they will handle the human resourcing.

That's why we use APIs!

Outsourcing your puppetry actor needs is like using an API.

A plugin of data that is managed externally.

Pls note: APIs don't have to be managed by an external company, it could be a different team who makes the API. But the important thing, is that it is *decoupled* from your website.

Let's recap the differences data sources

**Local datasource
or Hard-coded**

- Static & Localised

Easy for a small scope that doesn't change often, or perhaps only one person who maintains it

vs

**APIs
or Microservice**

- Dynamic & Decoupled

Easy to scale and decentralise changes or updates

Example using an actual website

So, I made this clusterf of a thing (my website).

What you see, the images, the orange accents, the search box, everything is what you would call the '**front-end**'.

Words you'd hear from the jungle are:

- HTML
- CSS
- Javascript

The screenshot shows a website layout. At the top left is a circular profile picture of a woman. Next to it, the text "G'day" is written in a stylized font, followed by "My name is Sophia". To the right of this are four navigation links: "HOME", "ABOUT", "WRITING", and "MEDIA". Below these are two horizontal lines. Underneath the first line is a "Search" button with the word "Search..." next to it. Underneath the second line is a "Tags" section with the text "Customer Experi...". At the bottom of this sidebar are the words "digital strategy facilitation".

The main content area has a header "Sophee" and a sub-header "Welcome to my website! Below you will find some featured projects.". It contains two cards:

- Customer Experience Strategy**: A card with a photo of a group of people and the text "Join our global team. We're always looking for volunteers - see our open positions here". Below the photo is a "Benefits of working with us" section showing three small images.
- Design Thinking**: A card with a grid of information:

Job Requirements	Manage teams of remote volunteers	Skills They Need to Do Their Job
Name	Global Coordinator	Tools Workplace, Google Sheets, Google Docs
Age	18 - 30 years	How Their Job Is Measured By
Occupation	Full-time student	Research, Oral documents
		Goals or Objectives at SDGNN Vault
		Full passion in contributing to sustainable development
		Biggest Challenges
		Balancing school work with PL responsibilities

Below the grid is the text "HCD process to build a strategy to tackle operational efficiency at a nonprofit".

Where the data comes from

The actual information ('About'), etc. would come from the back-end.

Words you'd hear from the jungle are:

- Java
- Python
- Javascript lmao

The screenshot shows a website layout for a person named Sophia. At the top right, there's a navigation bar with links to HOME, ABOUT, WRITING, and MEDIA. Below this, a breadcrumb navigation shows 'Home / About'. The main content area has a title 'About' and a sub-section 'Who are you?'. It includes a bio: 'I'm a consultant in the Digital & Transformation space, helping clients with things like reimagining their product customer experience, creating IT strategies, and reinventing their operating models to fit their digital needs.' Another bio section below states: 'I studied Software Development, at school, and in the past have worked as an iOS, Automation, and Web developer, in the Transport and Cyber Security sectors.' A third bio section mentions: 'I also lead the global operations & networks at the nonprofit SDSN Youth, as an advocate for creating meaningful impact by utilising human-centred approaches to design and technology.' At the bottom, there's a call to action 'Connect with me on LinkedIn!' followed by a LinkedIn profile snippet for 'Sophia Huynh'.

We won't mention how my website is actually a static site and has no back-end....

Using an API for Instagram 1/2

I also have my Instagram feed on my website too.

See, rather than manually uploading each picture, I simply use Instagram's API to automatically pull each photo.

Photography

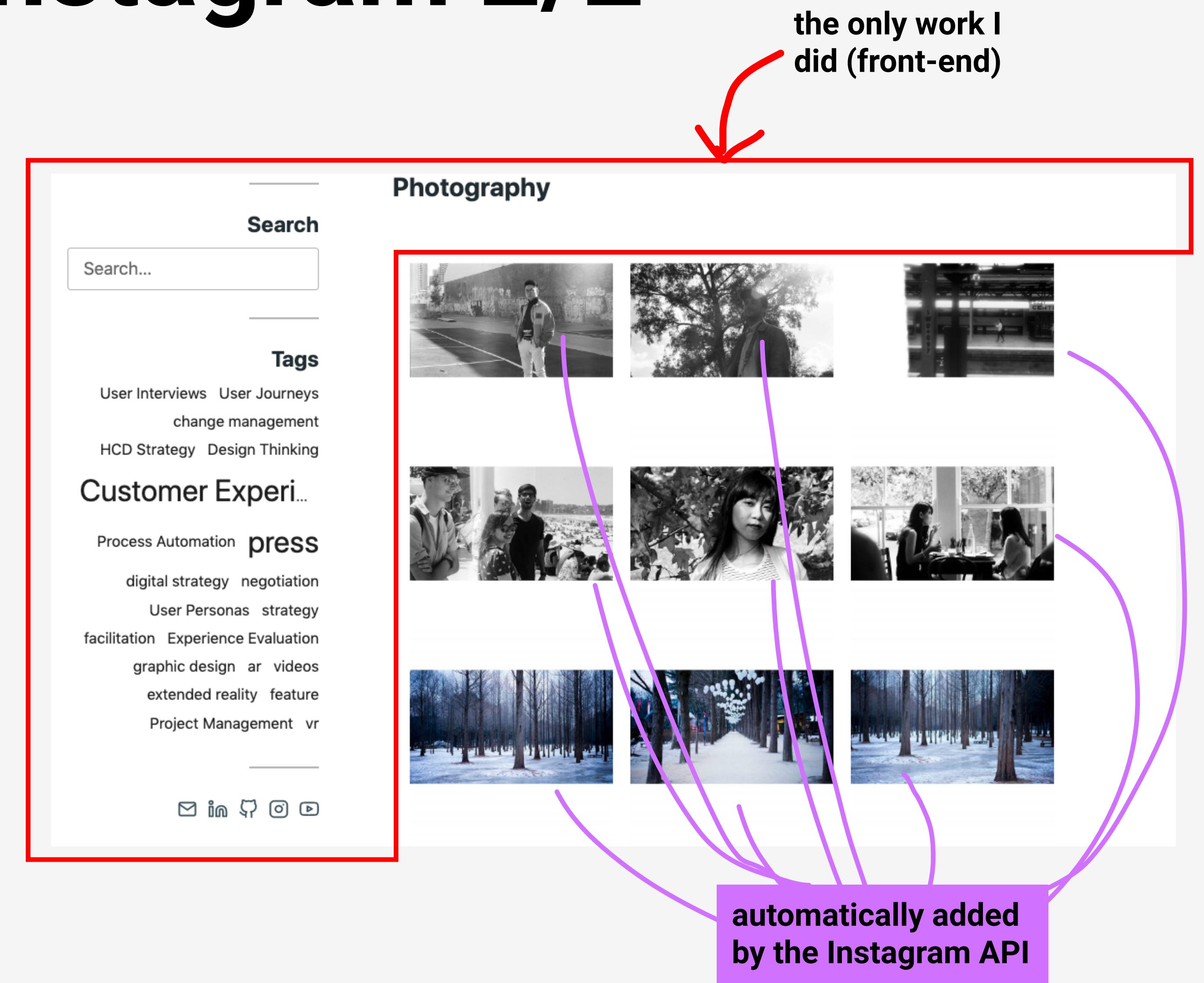


```
pe="text/javascript" src="https://soph.ee/js/instafeed.mi
="row">
instafeed.init({
  "url": "https://www.instagram.com/p/B-g0RPijHu7/",
  "url": "https://www.instagram.com/p/B-W1Pt5jsHy/",
  src="//scontent.cdninstagram.com/v/t51.2885-15/sh0.08/e3
c=EVCjtcU6MygAX9sBC6F&oh=bf0089b...&oe=5EB16885" == $0
  "url": "https://www.instagram.com/p/B-UcptRjKo2/",
  "url": "https://www.instagram.com/p/B-0-VWjDcjG/";
```

Using an API for Instagram 2/2

Whenever I upload a new Instagram post, I don't have to upload a new picture manually.

I just create the visual space for the photos (e.g. how far the images are from each other, that I want 3 images max in one row, etc) and the API will slot the images in the right place.



Thanks for reading shitman's layman version: how does a website work?

If you want to follow an account with angsty yet
nostalgic photography, follow me at @sph_ea



^ my fav fb sticker
Mobile Girl, Mim