devfest 2022

book-nav-toggle''
dden="" fixed="" aria-label="Hide
Hide side navigation"

# A deep dive into web development.

Victor Ndaba
Full stack web and mobile developer

Google Developer Groups
Nyeri

# Objectives of this talk

1. To explain the difference between the internet and the web
2. Help you understand why the web was created and how it has evolved
3. To examine the various web frameworks in existence and their pros and cons
4. To provide you with a deeper understanding of the internet and the web

devfest
2022

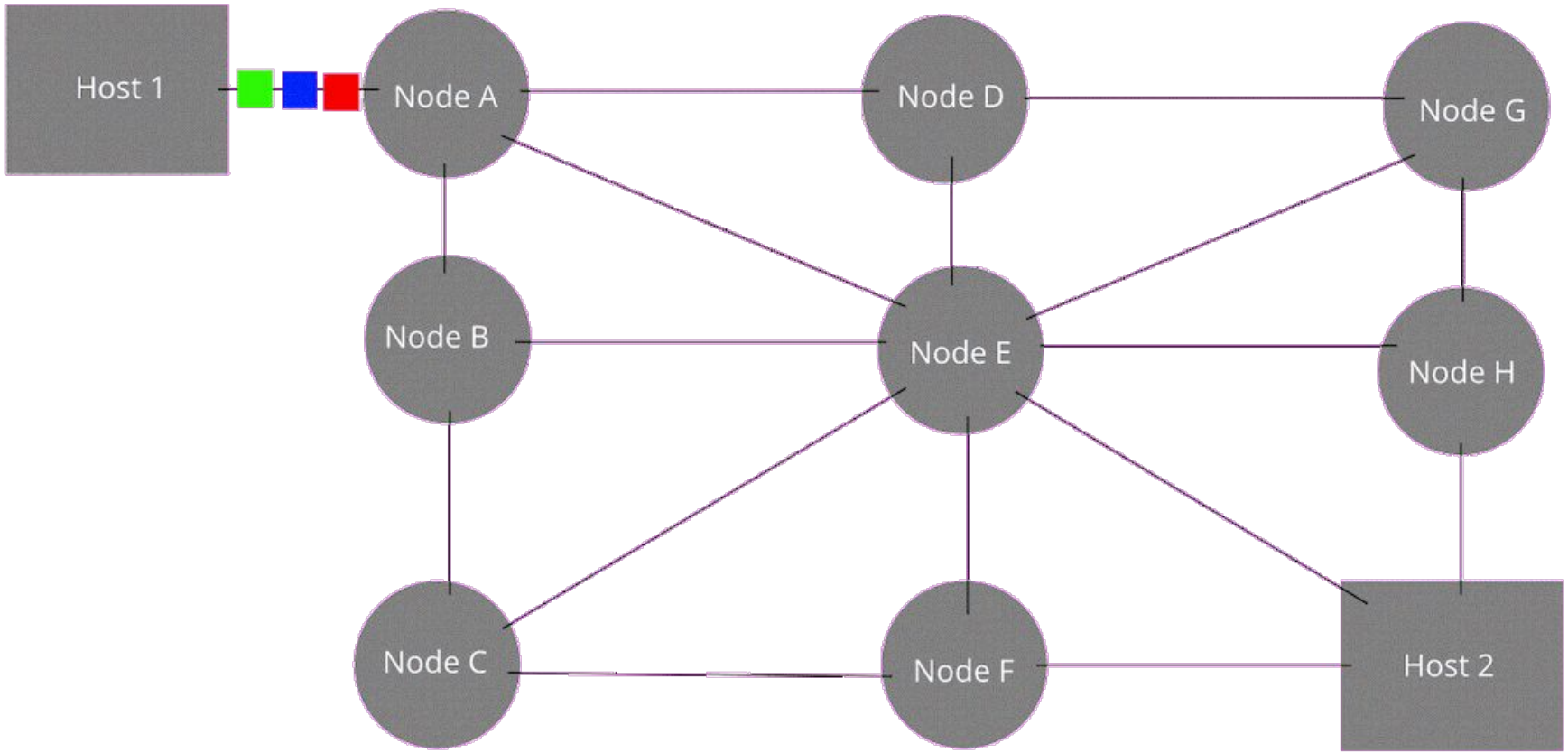
# The internet vs the web

# ARPANet (Advanced Research Projects Agency)

Before there was the internet, there was ARPANet. **ARPA** was a project launched by the **DOD** for projects that could be used for military advantage. Some sources say that ARPANet was started as an attempt by the **US govt** to create a **network** that could withstand a nuclear attack while others say that scientists created the network to improve time sharing systems.

It started as a connection between four universities in 1969 and the connections were made via telephone lines. To enable many computers to communicate simultaneously and prevent network congestion, the ARPANet Engineers came up with the concept of packet switching which is still in use to date.

The original message is Green, Blue, Red.

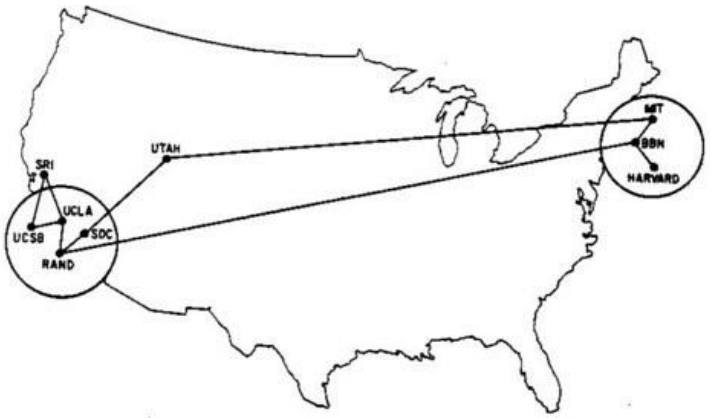


# ARPANet had some challenges...

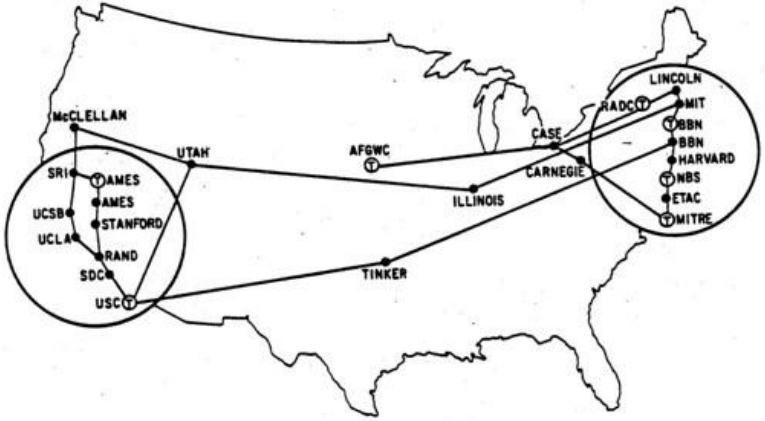
The packets sent via the network would contain a sequence no and the address of the computer it is headed to. To facilitate this, every node in the network would have a copy of an address table in order to move the packets along to the computer that is closest to the receiver. As the network grew, this made it difficult to scale. To solve this, in 1973, Stanford was chosen as the official record keeper for the addresses of every node in the network.

However, as ARPANet grew, the address record keeping and updating proved to be too much. Email packets dominated ARPANet in this days and to send an email, some users would have to trace out the path of the email packets manually. This was the beginning of the Domain Name System(DNS). Different computers on the network would store the addresses of different domains.

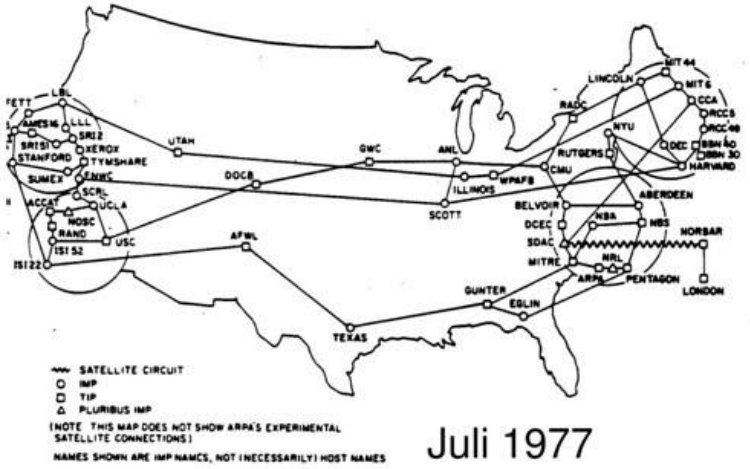

Dezember 1969




Juni 1970




März 1972




Juli 1977

# NPL + CYCLADES + ARPANET

The US, however, was not the only country developing such a network. Other similar networks existed in London and France and eventually, there were incorporated into ARPANet forming the basis of what is known today as the Internet.

However, each of these networks formatted their packets differently and this made it difficult for them to work together. This led to the creation of a standard protocol known as TCP/IP which is still used to date. TCP specified how the packets should be formatted and IP was a standard for how computer addresses should be formatted.

And just like that, the Internet was born. But it was still far from the internet we know of today and there was still one piece of the puzzle left.

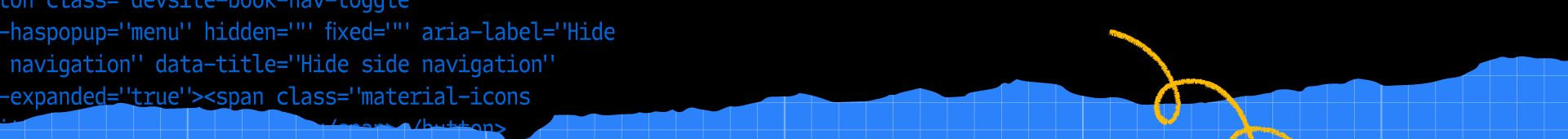
# The World Wide Web - the final piece

In the early days of the internet, it was only used by universities, govts and nerds who knew how to use special programs like usenet and gophernet to access information from the internet. Accessing resources such as files on from computers on different parts of the network was difficult.

At this time, information from the internet was displayed on terminal UIs and was only text-based. This was until 1989 when Sir Tim-Berners Lee combined the concepts of Hypertext, HTTP and URI(URL) to come up with what he termed the world wide web.

With this, it was now possible for anyone to connect to the internet. Information was now displayed via websites that could be easily accesed


Source: https://submarinecablemap.com

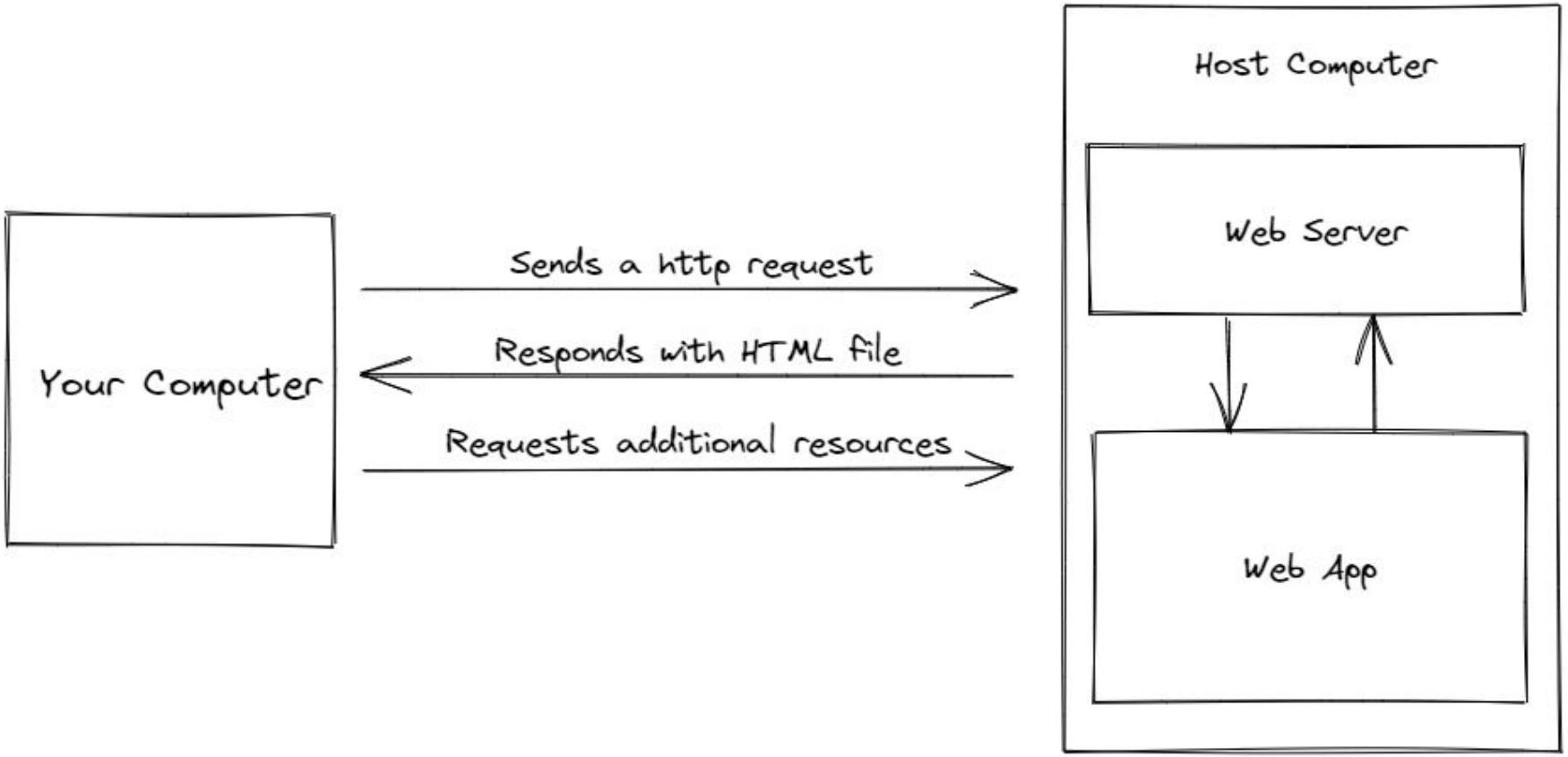



The internet is a physical network but the web is a platform allowing sharing of resources via the internet.

devfest
2022

# HTML and web development basics

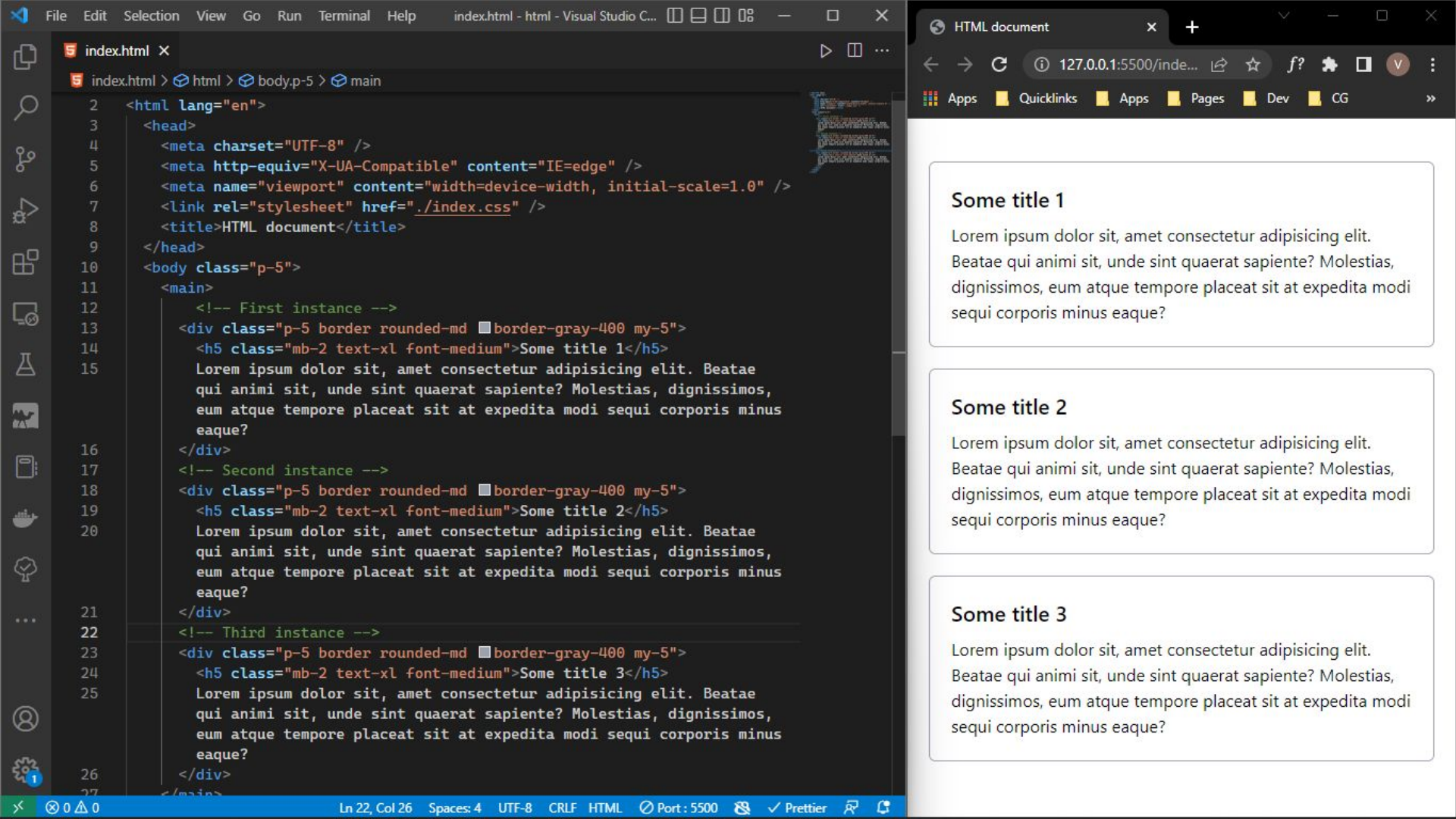
# Anatomy of a web request



# The problem with plain HTML - it doesn’t scale

Using plain HTML to build a complex website/web app is quite a nightmare. You end up constantly repeating similar blocks of markup which violates one of the golden rules of programming. **DRY(Don’t repeat yourself)**. HTML tightly binds the markup and data which makes it difficult to scale.

Web developers realized this very early on and sought out ways to make HTML more “scalable”. For our purposes today, we will look into two of these ways: using **web frameworks** which make our lives easier, or using a **CMS(Content Management System)** to separate data from markup.


File Edit Selection View Go Run Terminal Help
index.html - html - Visual Studio C...
index.html
index.html > html > body.p-5 > main
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta http-equiv="X-UA-Compatible" content="IE=edge" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<link rel="stylesheet" href="./index.css" />
<title>HTML document</title>
</head>
<body class="p-5">
<main>
<!-- First instance -->

<h5 class="mb-2 text-xl font-medium">Some title 1</h5>
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?

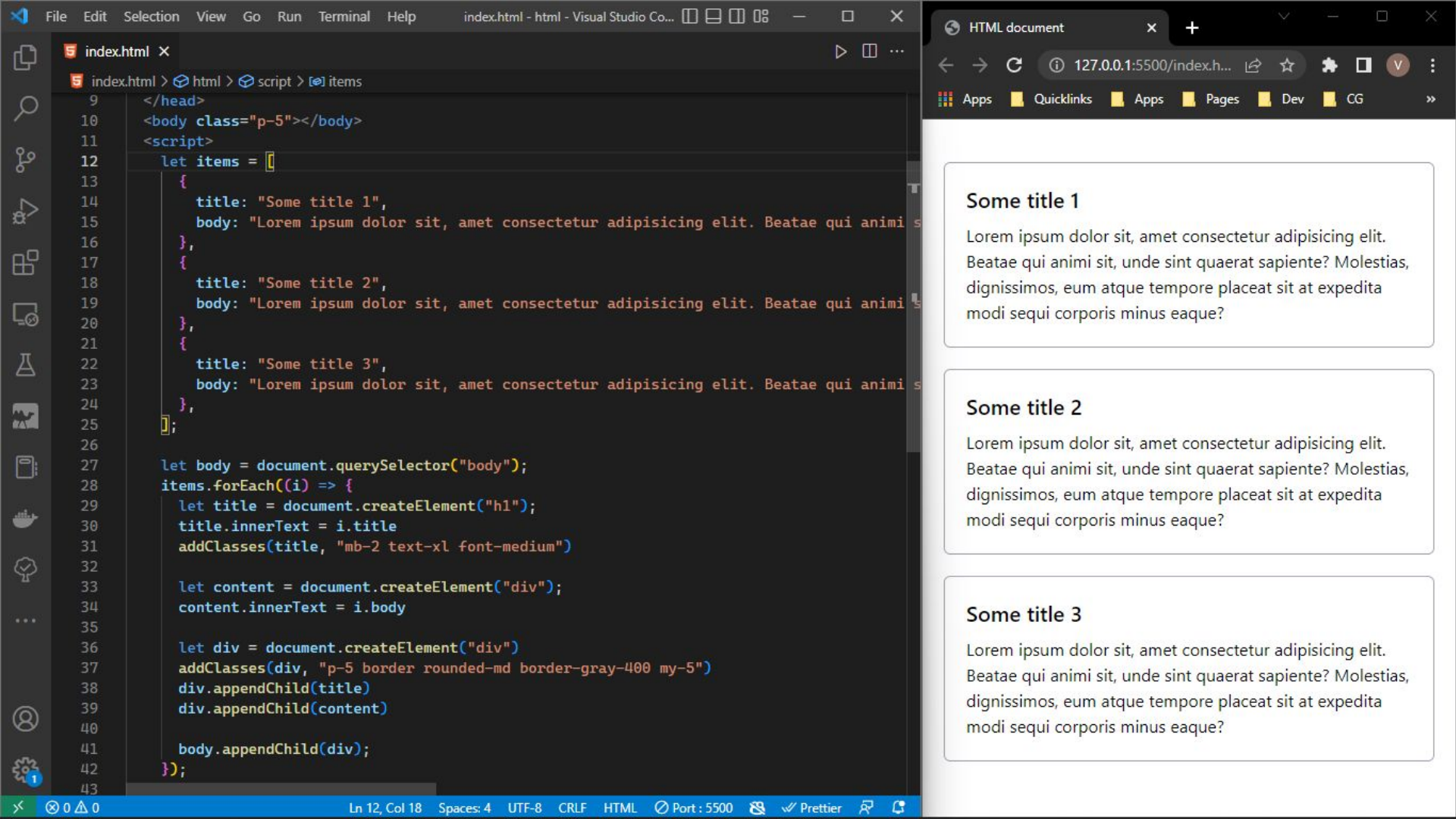
<!-- Second instance -->

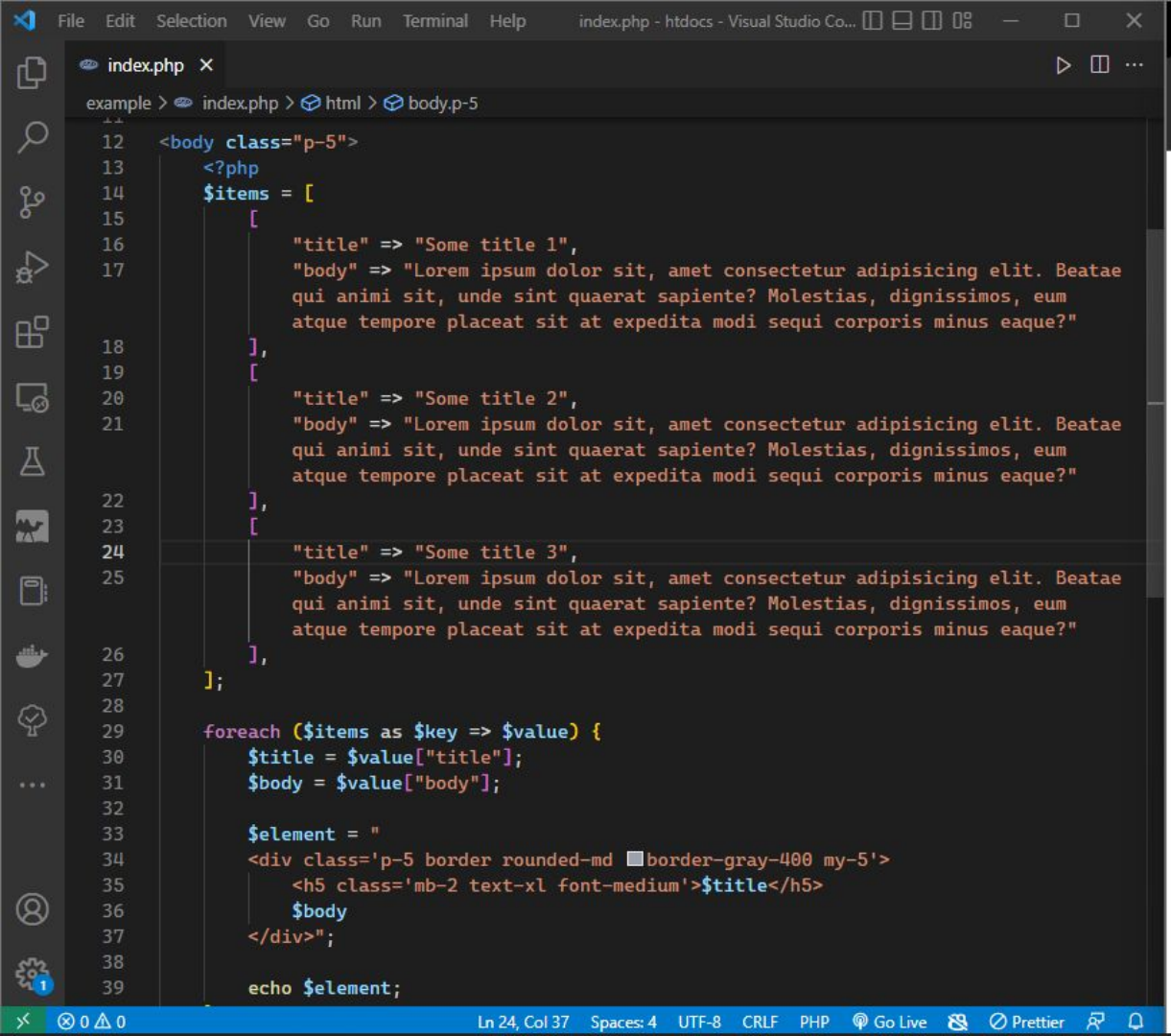
<h5 class="mb-2 text-xl font-medium">Some title 2</h5>
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?

<!-- Third instance -->

<h5 class="mb-2 text-xl font-medium">Some title 3</h5>
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?

Ln 22, Col 26 Spaces: 4 UTF-8 CRLF HTML Port : 5500 Prettier
HTML document
127.0.0.1:5500/inde...
Apps Quicklinks Apps Pages Dev CG
Some title 1
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 2
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 3
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?


File Edit Selection View Go Run Terminal Help
index.html - html - Visual Studio Co...
index.html
index.html > html > script > items
</head>
<body class="p-5"></body>
<script>
let items = [
{
title: "Some title 1",
body: "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi s
},
{
title: "Some title 2",
body: "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi s
},
{
title: "Some title 3",
body: "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi s
},
];
let body = document.querySelector("body");
items.forEach((i) => {
let title = document.createElement("h1");
title.innerText = i.title
addClasses(title, "mb-2 text-xl font-medium")
let content = document.createElement("div");
content.innerText = i.body
let div = document.createElement("div")
addClasses(div, "p-5 border rounded-md border-gray-400 my-5")
div.appendChild(title)
div.appendChild(content)
body.appendChild(div);
});
Ln 12, Col 18 Spaces: 4 UTF-8 CRLF HTML Port : 5500 Prettier
HTML document
127.0.0.1:5500/index.h...
Apps Quicklinks Apps Pages Dev CG
Some title 1
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 2
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 3
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?


File Edit Selection View Go Run Terminal Help
index.php - htdocs - Visual Studio Co...
index.php
example > index.php > html > body.p-5
<body class="p-5">
<?php
$items = [
[
"title" => "Some title 1",
"body" => "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?"
],
[
"title" => "Some title 2",
"body" => "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?"
],
[
"title" => "Some title 3",
"body" => "Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?"
],
];
foreach ($items as $key => $value) {
$title = $value["title"];
$body = $value["body"];
$element = "

<h5 class='mb-2 text-xl font-medium'>$title</h5>
$body
";
echo $element;
Ln 24, Col 37 Spaces: 4 UTF-8 CRLF PHP Go Live Prettier
Document
localhost/example/ind...
Apps Quicklinks Apps Pages Dev CG
Some title 1
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 2
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?
Some title 3
Lorem ipsum dolor sit, amet consectetur adipisicing elit. Beatae qui animi sit, unde sint quaerat sapiente? Molestias, dignissimos, eum atque tempore placeat sit at expedita modi sequi corporis minus eaque?

devfest
2022

# Let’s talk about wordpress

Wordpress powers

# 35%

Of the web

# What are Content Management Systems (CMS')?

- As we've discussed above, tight coupling between the HTML Markup and data does not scale
- CMS' like wordpress help separate the data and the markup. Most of these CMS' come with predefined themes allowing you to simply select a theme then simply edit the data which is stored in a database.
- However, the problem with wordpress,it is not flexible. i.e. you can't use a framework like react with wordpress. In recent years, Headless CMS' have been developed to solve this problem.

devfest
2022

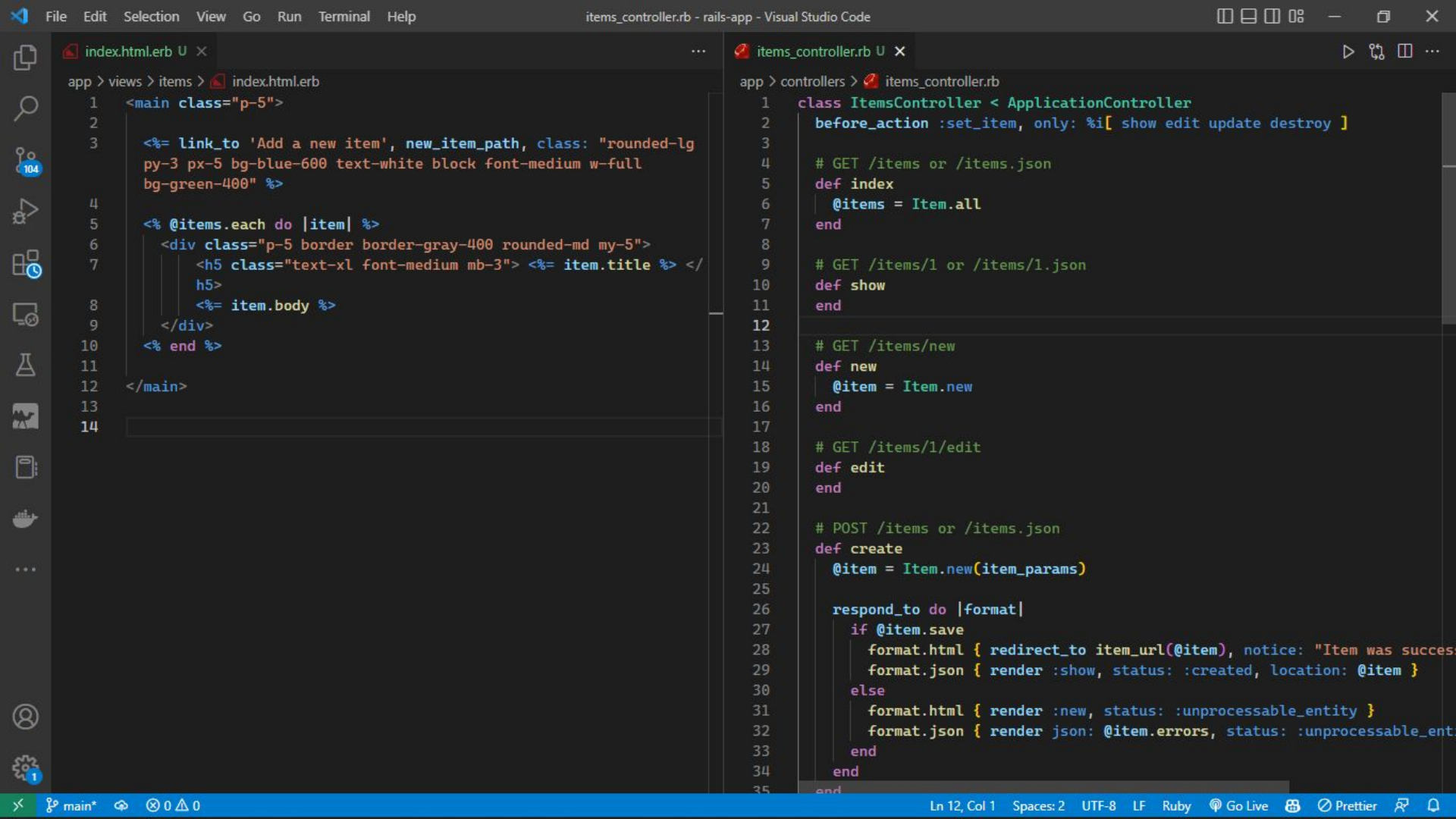
# Web frameworks

# What does a (fullstack) web framework do?

- Provide an abstraction over your database by providing you with an Object Relational Mapper(ORM) that translates code in your preferred language to database queries - **Model**
- Provide a routing system to help you map a url in the browser to the code you want executed when the url is hit - **Controller**
- Provide a way to directly insert data from your database directly into your HTML usually via a templating language. - **View**

# Ruby on Rails - Ruby


- Ruby on rails was created in July 2004 by DHH and went on to simplify fullstack web development.
- Rails follows the MVC architecture pattern and uses an ORM known as Active Record and has a very powerful CLI and works especially well with SQL databases.
- Rails is what is known as a batteries included framework meaning that it comes with a ton included out of the box.
- RoR used an embedded ruby(erb) syntax to enable you to render your data directly inside of HTML.


File Edit Selection View Go Run Terminal Help
items_controller.rb - rails-app - Visual Studio Code
index.html.erb U
app > views > items > index.html.erb
<main class="p-5">
<%= link_to 'Add a new item', new_item_path, class: "rounded-lg py-3 px-5 bg-blue-600 text-white block font-medium w-full bg-green-400" %>
<% @items.each do |item| %>

<h5 class="text-xl font-medium mb-3"> <%= item.title %> </h5>
<%= item.body %>

<% end %>
</main>
items_controller.rb U
app > controllers > items_controller.rb
class ItemsController < ApplicationController
before_action :set_item, only: %i[ show edit update destroy ]
# GET /items or /items.json
def index
@items = Item.all
end
# GET /items/1 or /items/1.json
def show
end
# GET /items/new
def new
@item = Item.new
end
# GET /items/1/edit
def edit
end
# POST /items or /items.json
def create
@item = Item.new(item_params)
respond_to do |format|
if @item.save
format.html { redirect_to item_url(@item), notice: "Item was succes
format.json { render :show, status: :created, location: @item }
else
format.html { render :new, status: :unprocessable_entity }
format.json { render json: @item.errors, status: :unprocessable_ent
end
end
end
main*
0 0
Ln 12, Col 1
Spaces: 2
UTF-8
LF
Ruby
Go Live
Prettier

# Django - Python


- Django was created in July 2005 and has come to be loved for its simple yet powerful nature.
- It uses a Model-View-Template pattern which is generally similar to the MVC but with minor differences.
- Django was used to build sites like **`Spotify, Bitbucket, Youtube`**
- Django also comes with an Admin Dashboard which allows you to create or update data via a user interface.
- For templating, django has its own templating system built on top of HTML that extends the functionality.

File Edit Selection View Go Run Terminal Help

views.py - django-app - Visual Studio Code

items.html

```
<main>
{% comment %} Loop through items and display {% endcomment %}
    {% for item in items %}
        <div class="item">
            <h2>{{ item.title }}</h2>
            <p>{{ item.body }}</p>
        </div>
    {% endfor %}
</main>
```

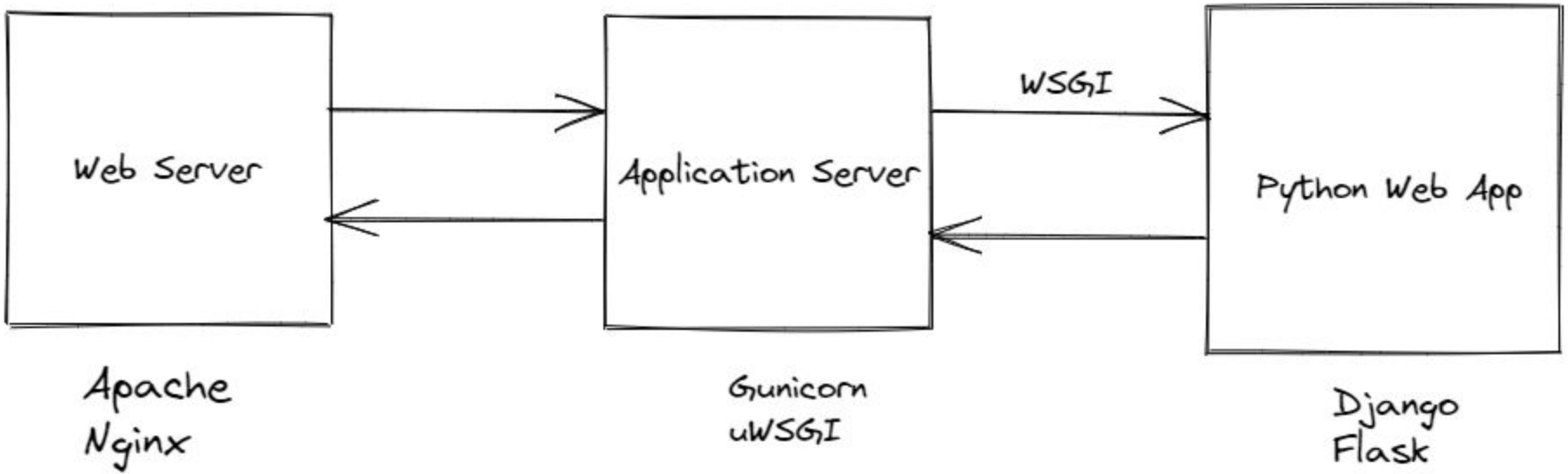
views.py

```
from django.shortcuts import render
from .models import Item

# Create your views here.

def get_items(request):
    """ A view to return the items page """
    items = Item.objects.all()
    context = {
        'items': items,
    }
    return render(request, 'items/items.html', context)
```

Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.10.5 ('env': venv) Go Live Prettier

# More on python frameworks...

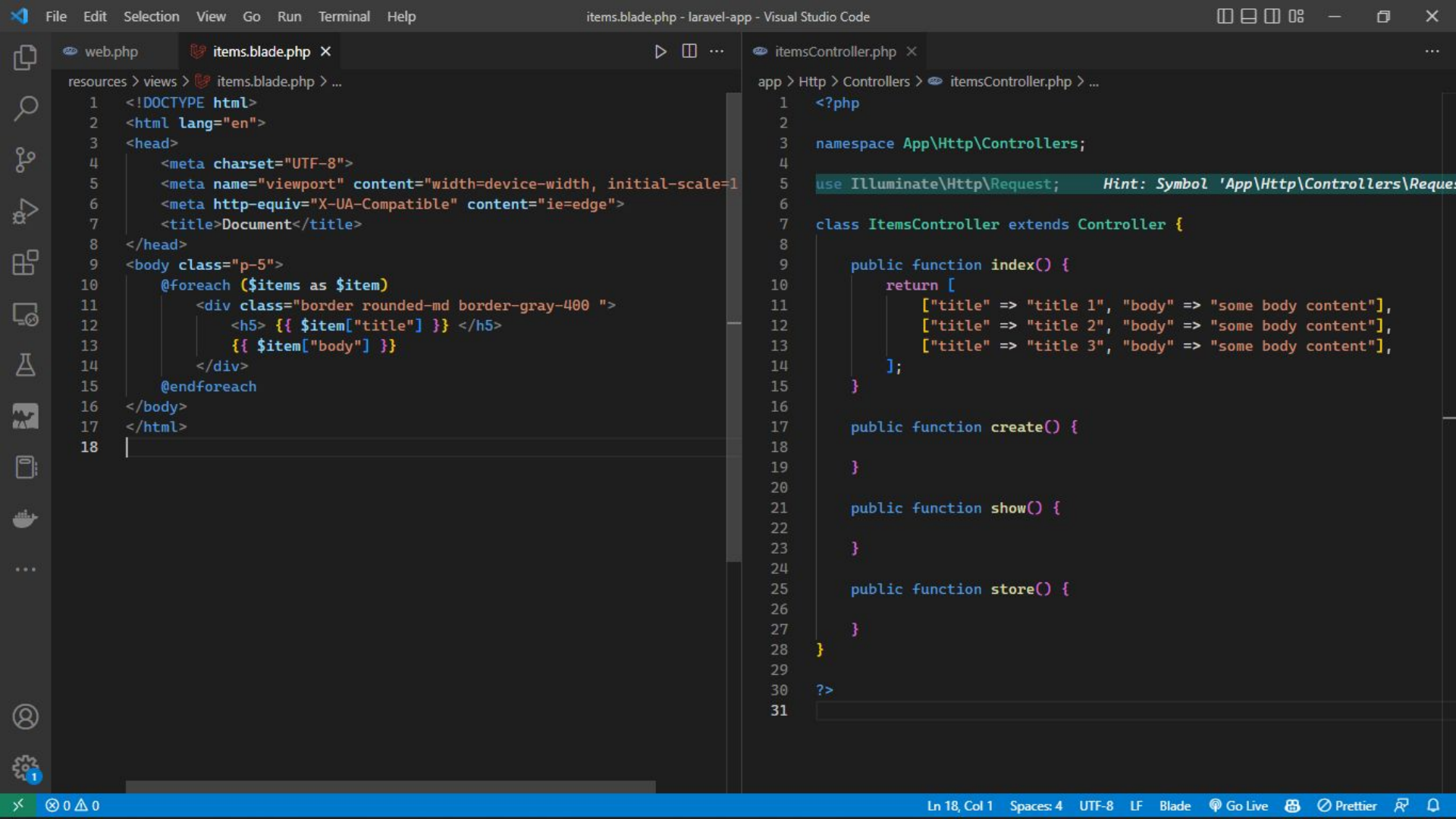
- In production, python apps created using frameworks like django or flask need additional components to be run. One such component is known as Gunicorn which is used as an interface between a web server and app.



# Laravel - php

Laravel

- Php alias hypertext preprocessor alias personal homepage was the original server side language for the web. Laravel is a framework that builds on top of php providing more features with less hassle.
- Laravel was created in **June 2011** and is the most popular php framework. It is a batteries included kind of framework and has robust tools to help you speed up development
- For templating, laravel builds on top of php with **blade templates** which allow you to write dynamic php in your html.
- It also follows a **model view controller pattern** and has a reasonably large community.
- Laravel has a tool called **inertia** that enables you to use a framework like vue to declare your frontend UI but still use laravel.


File Edit Selection View Go Run Terminal Help
items.blade.php - laravel-app - Visual Studio Code
web.php
items.blade.php
resources > views > items.blade.php > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="UTF-8">
5 <meta name="viewport" content="width=device-width, initial-scale=1
6 <meta http-equiv="X-UA-Compatible" content="ie=edge">
7 <title>Document</title>
8 </head>
9 <body class="p-5">
10 @foreach ($items as $item)
11 
12 <h5> {{ $item["title"] }} </h5>
13 {{ $item["body"] }}
14 
15 @endforeach
16 </body>
17 </html>
18
itemsController.php
app > Http > Controllers > itemsController.php > ...
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request; Hint: Symbol 'App\Http\Controllers\Reque
6
7 class ItemsController extends Controller {
8
9 public function index() {
10 return [
11 ["title" => "title 1", "body" => "some body content"],
12 ["title" => "title 2", "body" => "some body content"],
13 ["title" => "title 3", "body" => "some body content"],
14 ];
15 }
16
17 public function create() {
18
19 }
20
21 public function show() {
22
23 }
24
25 public function store() {
26
27 }
28 }
29
30 ?>
31
Ln 18, Col 1 Spaces: 4 UTF-8 LF Blade Go Live Prettier

# Spring boot - Java


- The spring framework was created in October 2002 and is a popular choice for building performant server side apps.
- Spring boot uses the strong OOP principles of the Java Language to create controllers that map data from models, which are also classes, into views. It is also an MVC framework.
- Plain Java can be used to serve HTML with emebedded Java code using **JSP(Java server pages)** and **servlets** but spring can take this a step further with **ThymeLeaf templates.**
- Spring benefits from the mature and well-developed Java ecosystem and is often the goto choice when building microservices.

# ASP.NET - C#

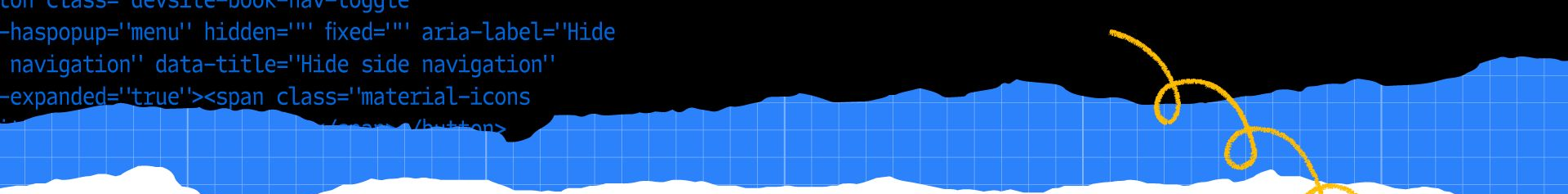

- The ASP framework was created by Microsoft in February 2002 from the C# language which brings together the strong OOP principles found in Java with the performance of C and it's really easy to use.
- The framework also features an MVC pattern and has matured nicely over the years. It's one of the most popular frameworks due to its simple yet robust nature.
- Views in ASP are declared in .cshtml files which allow dynamic C# code inside of your HTML document.
- It uses Eloquent ORM to abstract database functionality.

# Honorable mentions....

- **Phoenix Framework (Elixir)** - It is very similar to ruby in both syntax as well as architecture but it uses Elixir which compiles down to Erlang bytecode which it makes it very performant.
- **Actix Web (Rust)** - Rust doesn't have a dominant framework in the ecosystem but one of the popular ones is actix web which is a used to make blazingly fast web apps.
- **Gin Framework (Golang)** - Golang is an extremely fast language that comes with its own templating language which the Gin framework builds on top of to make development easier.

devfest
2022

# Javascript frameworks




Anything that can be written in JavaScript will eventually be written in JavaScript.

*~ Atwood's Law(Jeff Atwood)*

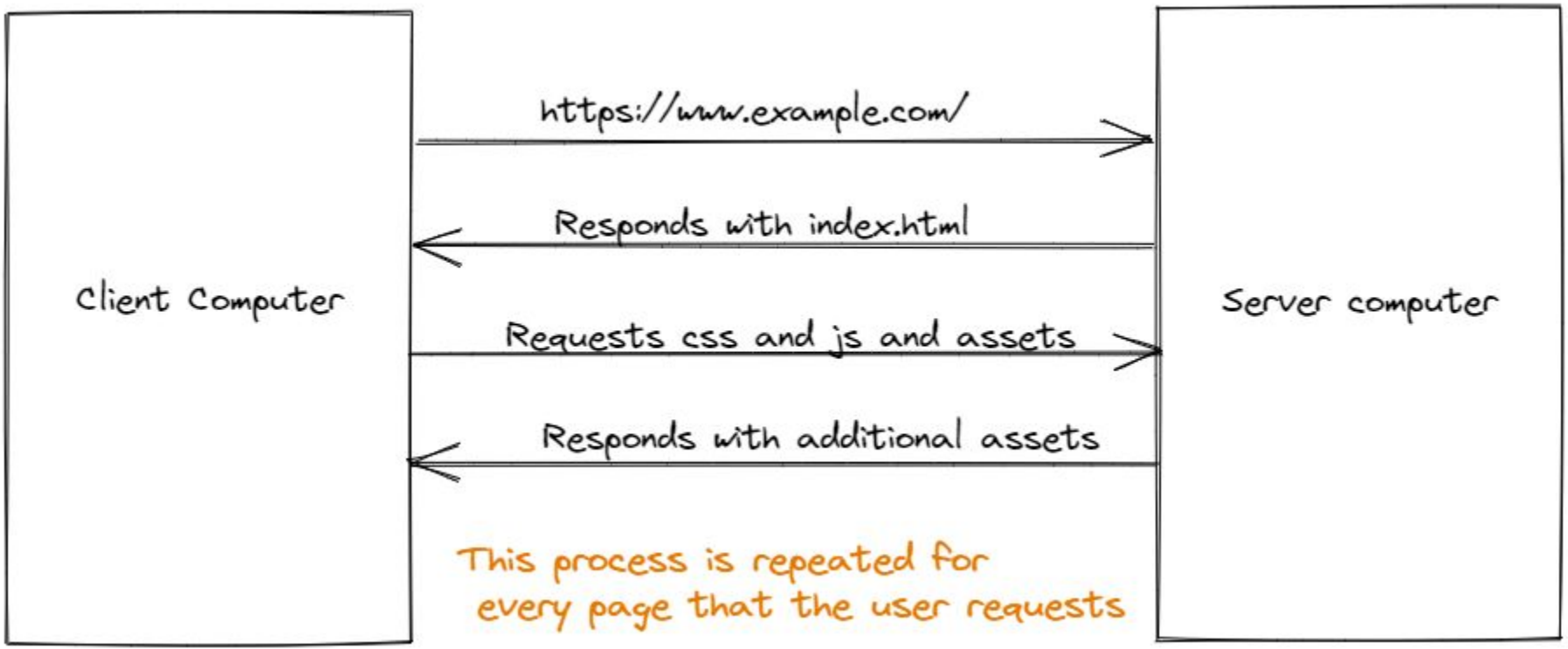
# The need for module bundlers

- Javascript was created to make forms more interactive. When the V8 engine parses JS in the browser, it only reads it from one file. It doesn't have a module system, ergo, it doesn't understand "import" statements.
- But during development, it is impossible to write everything in one single file. Developers want to split code across multiple files but the JS engine requires only one file. Module bundlers are used to reconcile this difference.
- But JS developers took this even further. Bundlers are now used to process multiple assets including CSS files and images etc. Examples are **Webpack, Vite, SWC, Snowpack, Parcel**

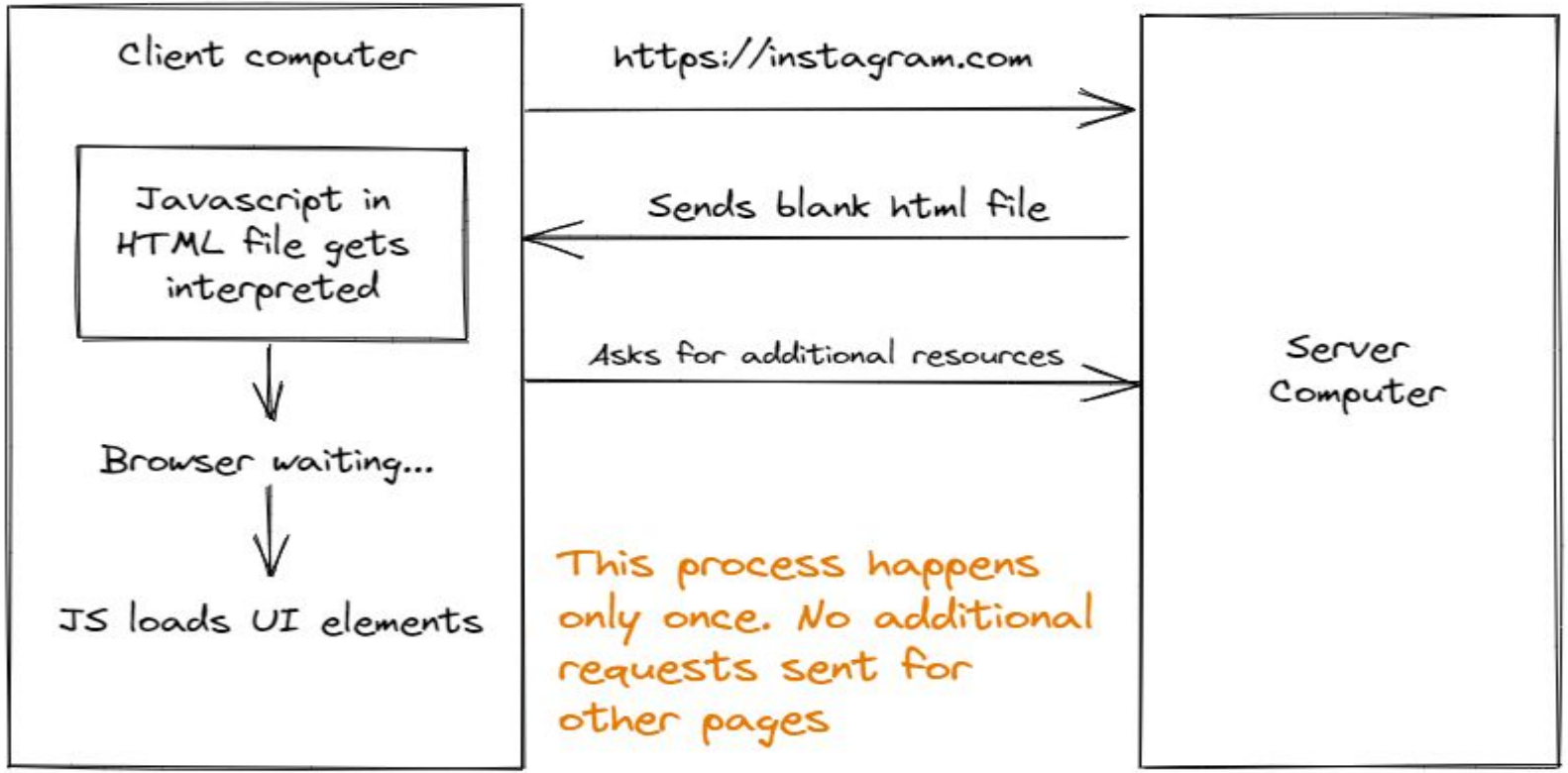
# JS Frameworks and Rendering Techniques

- In the JS ecosystem, the frameworks are usually categorized by how they handle rendering of pages:
- **Server Side Rendering** - This is how traditional web pages were served. Each page gets “rendered” everytime a new request is sent to the server. This option is usually preferred for websites that require strong SEO. Such apps are often called Multi-page apps
- **Client Side Rendering** - In recent years, Javascript has proved critical to web development. Frameworks that rely entirely on JS were created and were used to create Single Page Apps. Such apps only consist of a blank HTML page on initial page request which then gets populated with content once Javascript is loaded on the page
- **Static Site Generation -** Static sites are those that contain little to no dynamic content and have strong SEO. Many frameworks have been developed for creation of such websites and are often used for things like blogs.

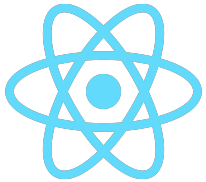
# Server Side Rendering



# Client Side Rendering



CSR(SPA)




SSR(MPA)



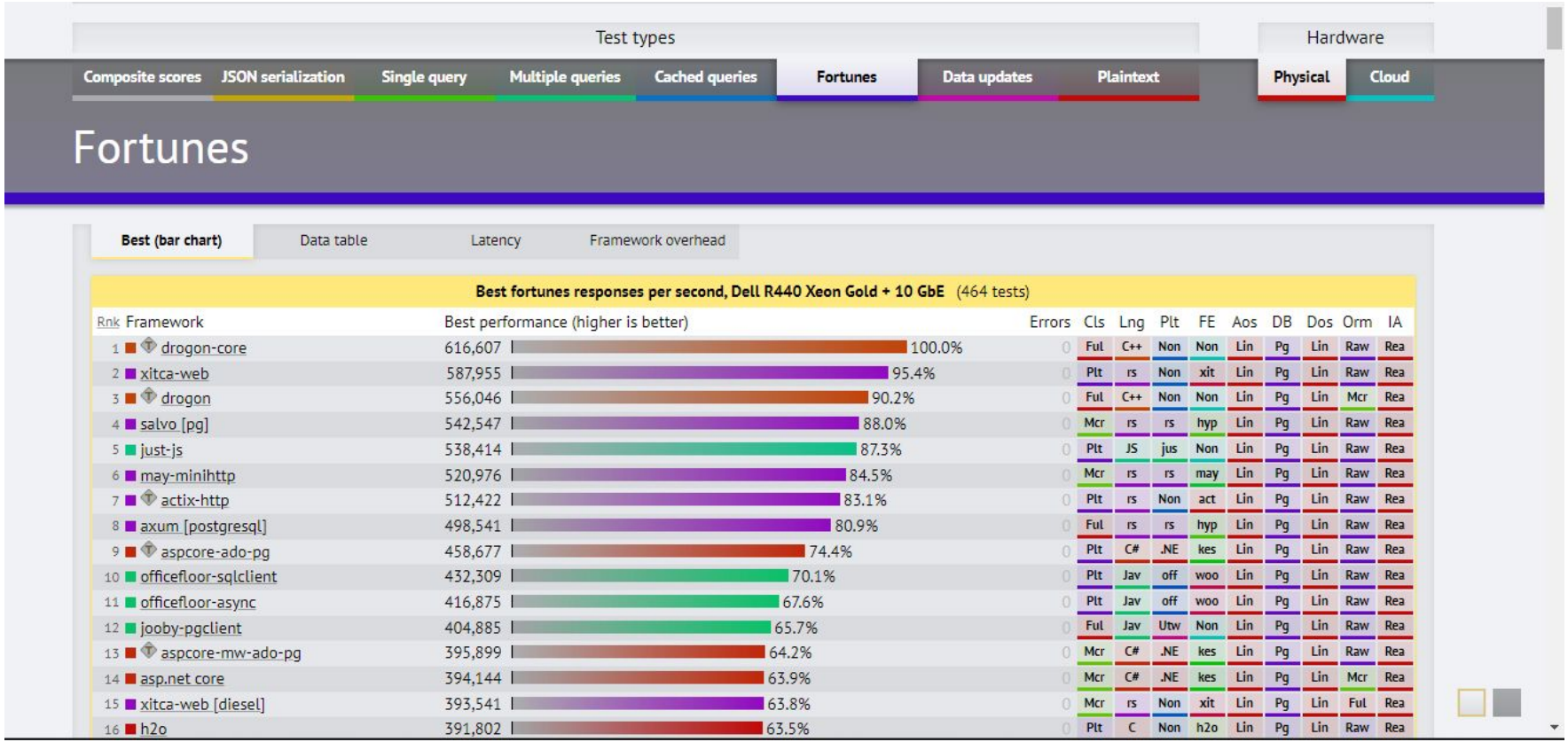
SSG(static)



devfest
2022

# Choosing a framework

# Speed (https://techempower.com/benchmarks)



Test types | Hardware

Composite scores | JSON serialization | Single query | Multiple queries | Cached queries | Fortunes | Data updates | Plaintext | Physical | Cloud

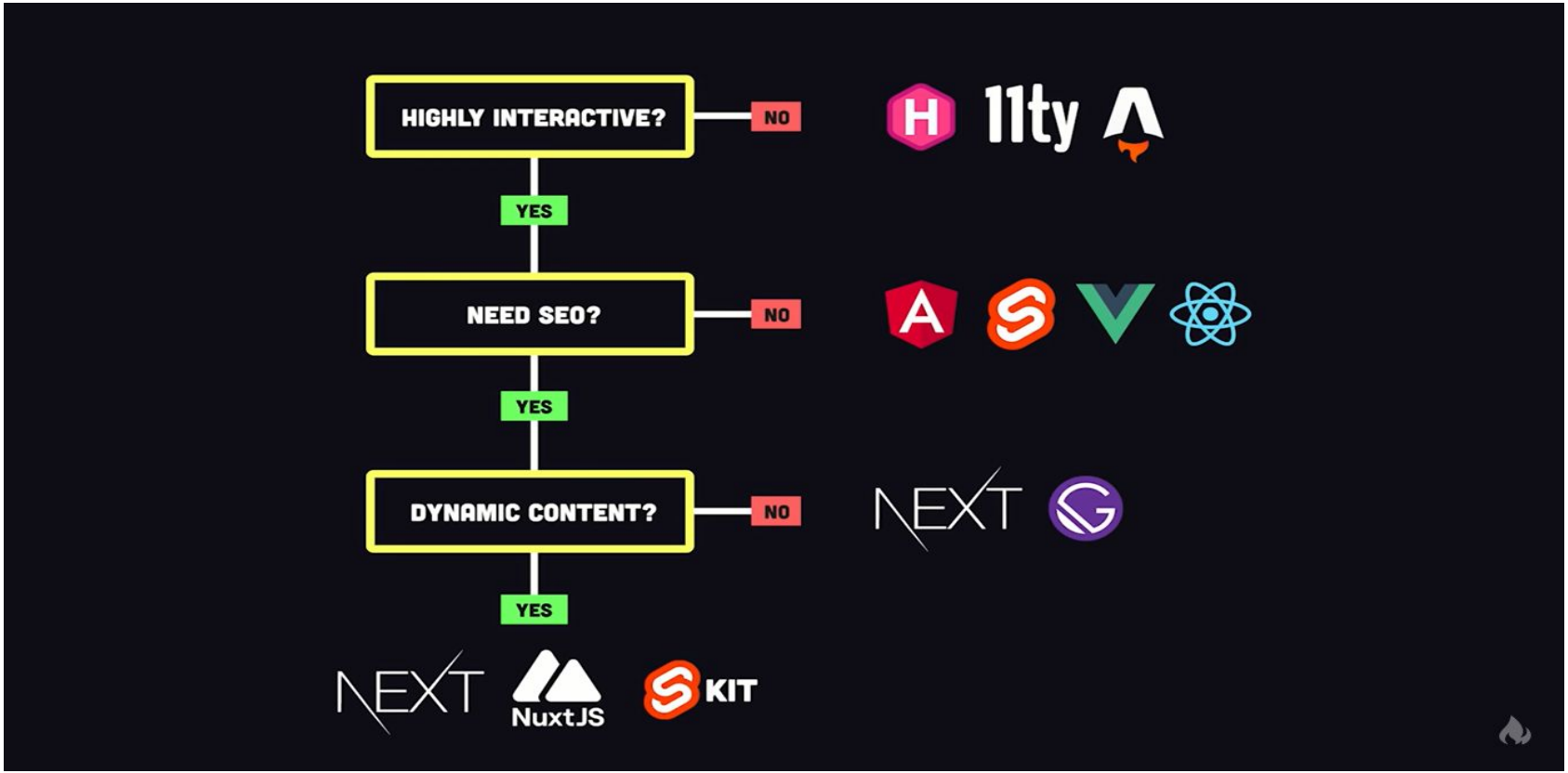
## Fortunes

Best (bar chart) | Data table | Latency | Framework overhead

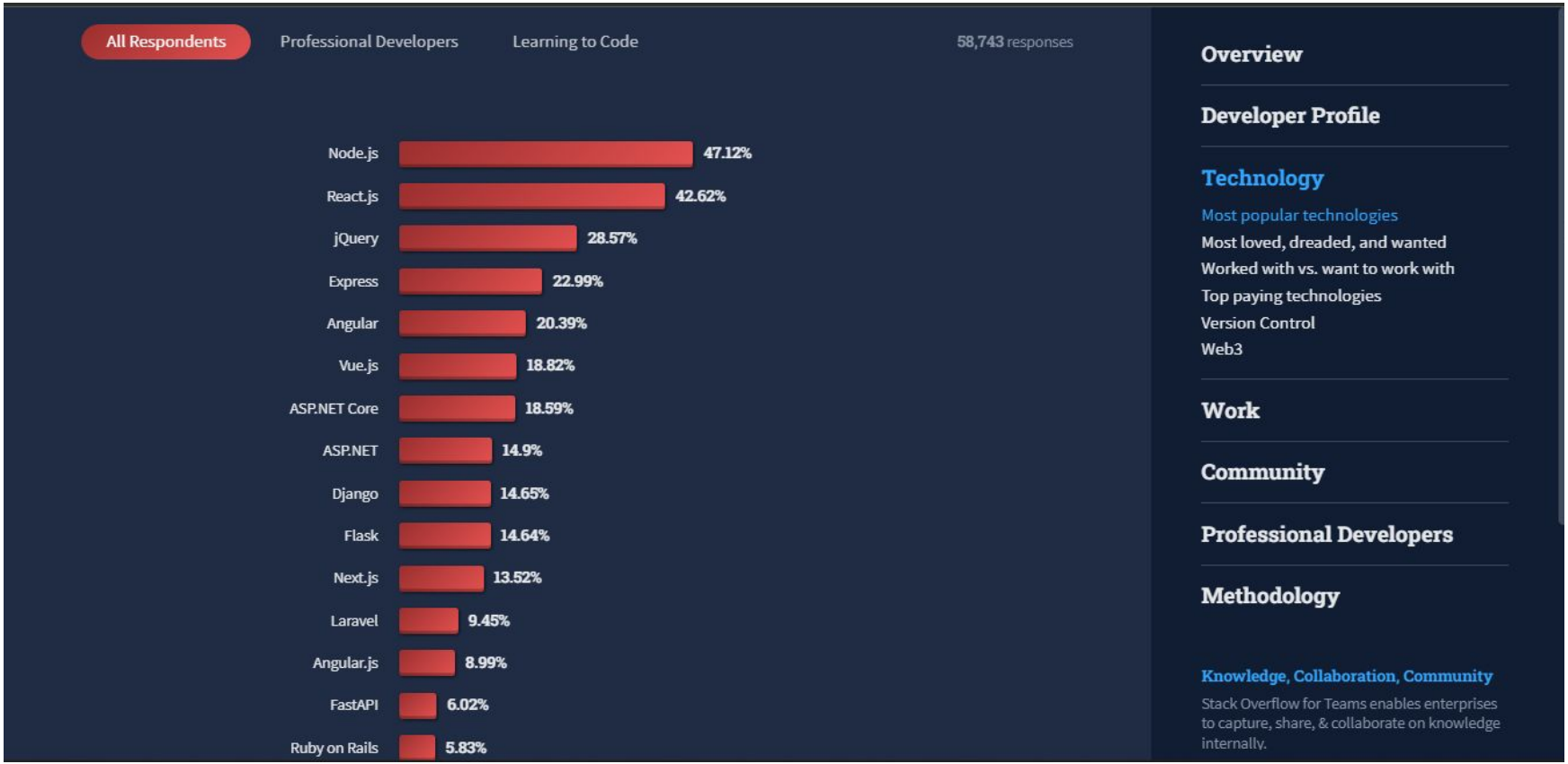
**Best fortunes responses per second, Dell R440 Xeon Gold + 10 GbE** (464 tests)

| Rnk | Framework | Best performance (higher is better) | | Errors | Cls | Lng | Plt | FE | Aos | DB | Dos | Orm | IA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | drogon-core | 616,607 | 100.0% | 0 | Ful | C++ | Non | Non | Lin | Pg | Lin | Raw | Rea |
| 2 | xitca-web | 587,955 | 95.4% | 0 | Plt | rs | Non | xit | Lin | Pg | Lin | Raw | Rea |
| 3 | drogon | 556,046 | 90.2% | 0 | Ful | C++ | Non | Non | Lin | Pg | Lin | Mcr | Rea |
| 4 | salvo [pg] | 542,547 | 88.0% | 0 | Mcr | rs | rs | hyp | Lin | Pg | Lin | Raw | Rea |
| 5 | just-js | 538,414 | 87.3% | 0 | Plt | JS | jus | Non | Lin | Pg | Lin | Raw | Rea |
| 6 | may-minihttp | 520,976 | 84.5% | 0 | Mcr | rs | rs | may | Lin | Pg | Lin | Raw | Rea |
| 7 | actix-http | 512,422 | 83.1% | 0 | Plt | rs | Non | act | Lin | Pg | Lin | Raw | Rea |
| 8 | axum [postgresql] | 498,541 | 80.9% | 0 | Ful | rs | rs | hyp | Lin | Pg | Lin | Raw | Rea |
| 9 | aspcore-ado-pg | 458,677 | 74.4% | 0 | Plt | C# | .NE | kes | Lin | Pg | Lin | Raw | Rea |
| 10 | officefloor-sqlclient | 432,309 | 70.1% | 0 | Plt | Jav | off | woo | Lin | Pg | Lin | Raw | Rea |
| 11 | officefloor-async | 416,875 | 67.6% | 0 | Plt | Jav | off | woo | Lin | Pg | Lin | Raw | Rea |
| 12 | jooby-pgclient | 404,885 | 65.7% | 0 | Ful | Jav | Utw | Non | Lin | Pg | Lin | Raw | Rea |
| 13 | aspcore-mw-ado-pg | 395,899 | 64.2% | 0 | Mcr | C# | .NE | kes | Lin | Pg | Lin | Raw | Rea |
| 14 | asp.net core | 394,144 | 63.9% | 0 | Mcr | C# | .NE | kes | Lin | Pg | Lin | Mcr | Rea |
| 15 | xitca-web [diesel] | 393,541 | 63.8% | 0 | Mcr | rs | Non | xit | Lin | Pg | Lin | Ful | Rea |
| 16 | h2o | 391,802 | 63.5% | 0 | Plt | C | Non | h2o | Lin | Pg | Lin | Raw | Rea |

# Rendering (https://fireship.io)

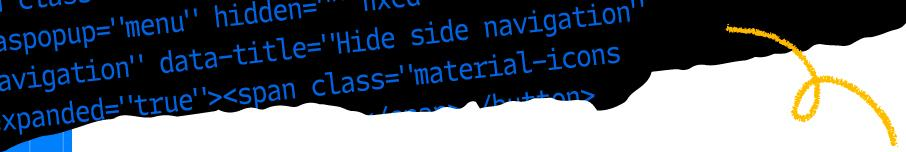


# Popularity (https://survey.stackoverflow.co/2022)



# Web 3 and the future of the web....

- The promise of a decentralized web is what Web 3.0 offers. Although exciting, Web 3.0 still has a long way to go and even if it achieves its goals, it would have to reinvent the whole web: it wouldn't be compatible with web 2.0
- Most architectures claiming to be 'Web 3' are not true to their claims. Perhaps, one of the few truly decentralized networks is https://pointnetwork.io
- In conclusion, the internet and the web are still pretty young; being 50 and 30yrs respectively. The future might be uncertain but one thing's for sure, the web isn't going anywhere anytime soon.




# THANK YOU

https://vndaba.rocks

mwangindaba@gmail.com

https://bit.ly/deepDiveIntoWebDev