Fullstack Development

CS571: Building User Interfaces

Warmup: Download today's starter code and install Docker on your computer!

Cole Nelson

Announcements

See "No Title" Announcement

What will we learn today?

- What is the software stack?
- How do we deploy a frontend?
- How can we use containerization?
- How can we develop a backend?
- How can we persist data?
- What are other considerations?

Software Stack

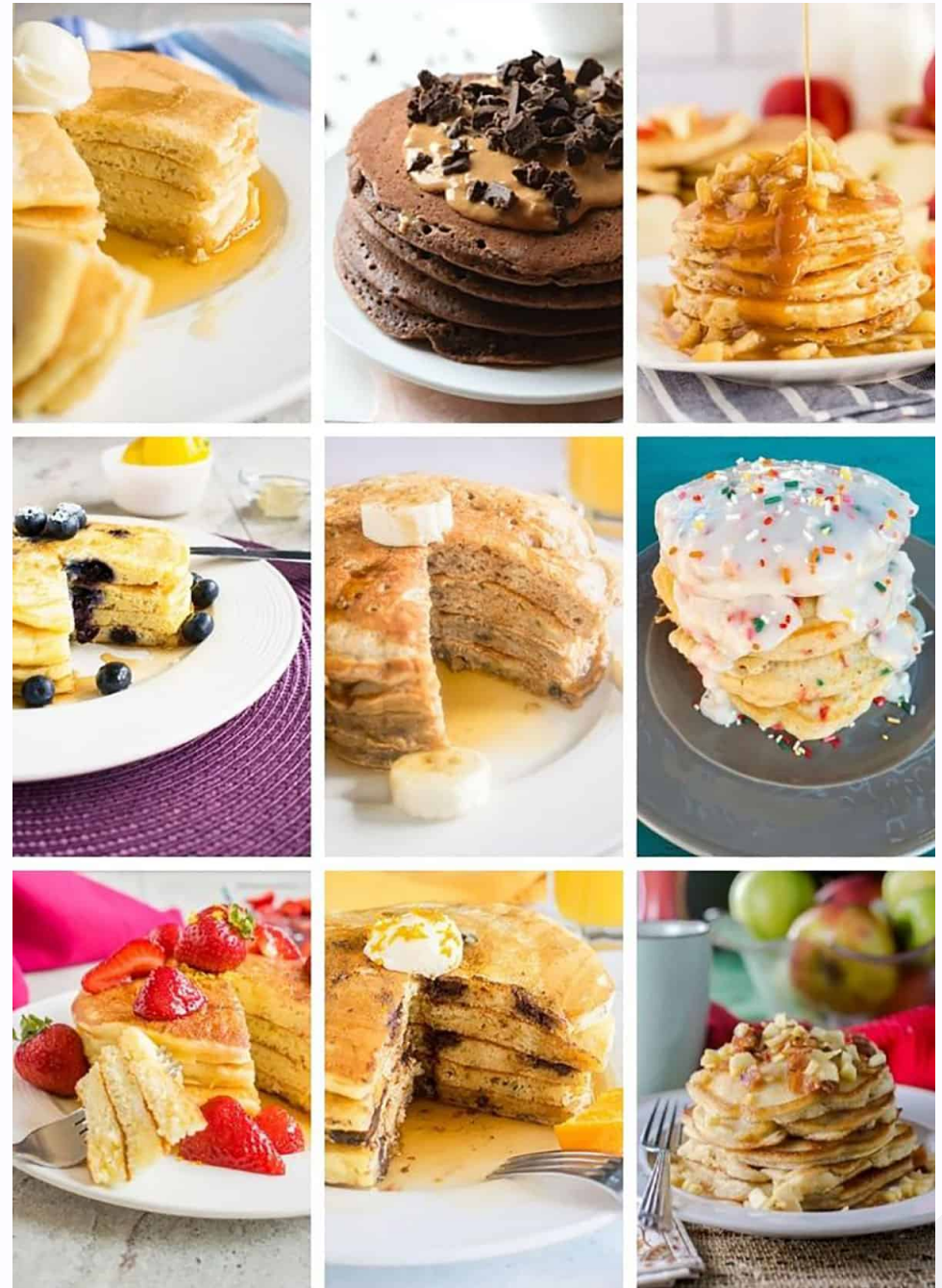Think of software like a stack of pancakes...

Image Source

Software Stack

... where each pancake can be its own flavor...

Image Source

Software Stack

... and can be cooked its own way...

Image Source

PANCAKES RECIPE

1. EGGS
2. FLOUR
3. MILK
4. SUGAR
5. COOKING OIL
6. SALT
7. BUTTER
8. FRYING PAN
9. WHISK
10. SPATULA
11. BOWL
12. CUP
13. SPOON

Software Stack

… with as many or as few as we want!

Image Source

The Browser

However, we are constrained to what the browser can interpret...

- HTML
- CSS
- JS

Delivery of React App

We don't deliver our JSX code, we deliver HTML, CSS, and JS generated via `npm run build`!

Also, specify a home page (absolute or relative)...

```json
{
    "name": "hw3",
    "version": "0.1.0",
    "private": true,
    "homepage": "https://coletnelson.us/mycoolapp/",
    "dependencies": {
        ...
    }
}
```

```html
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <link rel="icon" href="./favicon.ico" />
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <meta name="theme-color" content="#000000" />
    <meta name="description" content="Web site created using create-react-app" />
    <link rel="apple-touch-icon" href="./logo192.png" />
    <link rel="manifest" href="./manifest.json" />
    <title>React App</title>
    <script defer="defer" src="./static/js/main.aae268c3.js"></script>
    <link href="./static/css/main.ace4cd11.css" rel="stylesheet">
  </head>
  <body><noscript>You need to enable JavaScript to run this app.</noscript>
    <div id="root"></div>
  </body>
</html>
```

```
/*! For license information please see main.aae268c3.js.LICENSE.txt */
!function(){var e={694:function(e,t){var n;!function(){"use strict";var r={}.
hasOwnProperty;function l(){for(var e=[],t=0;t<arguments.length;t++){var n=arguments[t];
if(n){var a=typeof n;if("string"===a||"number"===a)e.push(n);else if(Array.isArray(n)){
if(n.length){var o=l.apply(null,n);o&&e.push(o)}}else if("object"===a)if(n.toString===
Object.prototype.toString)for(var u in n)r.call(n,u)&&n[u]&&e.push(u);else e.push(n.
toString())}}return e.join(" ")}e.exports?(l.default=l,e.exports=l):void 0===(n=function
(){return l}.apply(t,[]))||(e.exports=n)}()},618:function(e,t,n){var r;!function(){"use
strict";var l=!("undefined"===typeof window||!window.document||!window.document.
createElement),a={canUseDOM:l,canUseWorkers:"undefined"!==typeof Worker,
canUseEventListeners:l&&!(!window.addEventListener&&!window.attachEvent),canUseViewport:
l&&!!window.screen};void 0===(r=function(){return a}.call(t,n,t,e))||(e.exports=r)}()},
888:function(e,t,n){"use strict";var r=n(47);function l(){}function a(){}a.
resetWarningCache=l,e.exports=function(){function e(e,t,n,l,a,o){if(o!==r){var u=new
Error("Calling PropTypes validators directly is not supported by the `prop-types`
package. Use PropTypes.checkPropTypes() to call them. Read more at
http://fb.me/use-check-prop-types");throw u.name="Invariant Violation",u}}function t(){
return e}e.isRequired=e;var n={array:e,bigint:e,bool:e,func:e,number:e,object:e,string:e
,symbol:e,any:e,arrayOf:t,element:e,elementType:e,instanceOf:t,node:e,objectOf:t,oneOf:t
,oneOfType:t,shape:t,exact:t,checkPropTypes:a,resetWarningCache:l};return n.PropTypes=n,
n}},7:function(e,t,n){e.exports=n(888)()},47:function(e){"use strict";e.exports=
"SECRET_DO_NOT_PASS_THIS_OR_YOU_WILL_BE_FIRED"},463:function(e,t,n){"use strict";var r=n
(791),l=n(296);function a(e){for(var t=
```
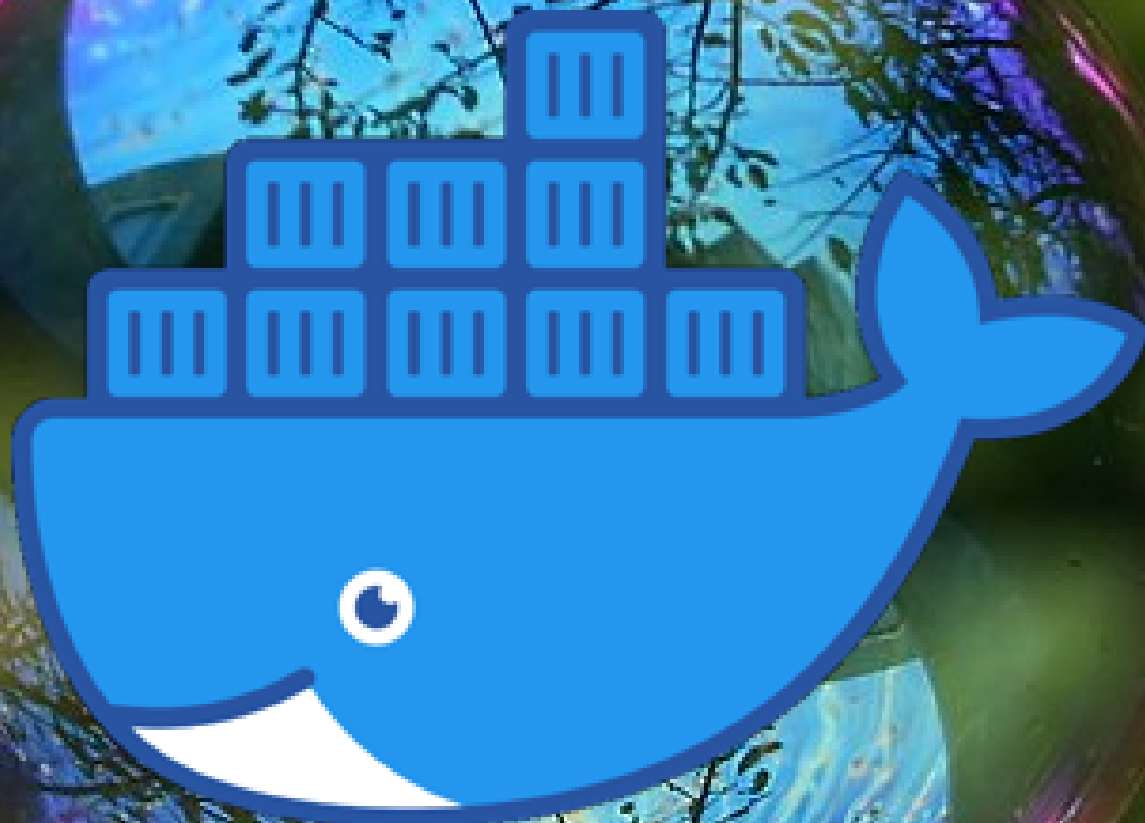
Build Bundle Deployment

Not an endorsement of any particular service.

Let's deploy cs571.org!

This is great but...

...dragging and dropping files?? Really??

Dockerfile → build → Docker Image → run → Docker Container

Image Source

Let's Build & Deploy
BadgerChat Basic

How to persist data?

Building a backend!

Creating a Backend Server

Many, many, many options!

- Google Cloud Functions
- AWS Lambdas
- C# & .NET
- Java & Spring
- Python & Flask
- **JavaScript & Express**

Let's Build an API!

Using JavaScript & Express
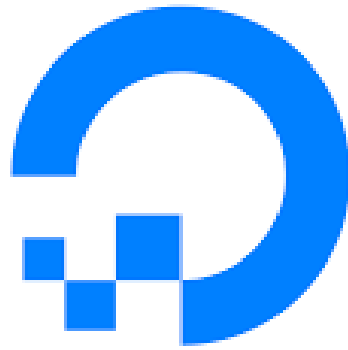
How to persist data?

Let's use SQLite

SQLite

- SQL, but lite!
- Creates a `.db` file on your machine
- Is not a "hosted" database, but is good for quick projects and hacks!
- Handles our concurrency issues.

Backend Server Hosting

Not an endorsement of any particular service.

Other Considerations

- Use Jenkins or some other CICD platform to create a build and deploy pipeline.
  - Include testing as an automated step.
- Use HTTPS for a secure HTTP connection.
  - Consider LetsEncrpyt.
- Buy a domain name?
  - Completely optional!

What's Next?

CS570 Introduction to Human-Computer Interaction
→ explore UX methods (research, design, evaluation)

CS770 Human-Computer Interaction
→ core topics and research methods in HCI research

See also hci.cs.wisc.edu.

What did we learn today?

- What is the software stack?
- How do we deploy a frontend?
- How can we use containerization?
- How can we develop a backend?
- How can we persist data?
- What are other considerations?

Thank you! :)