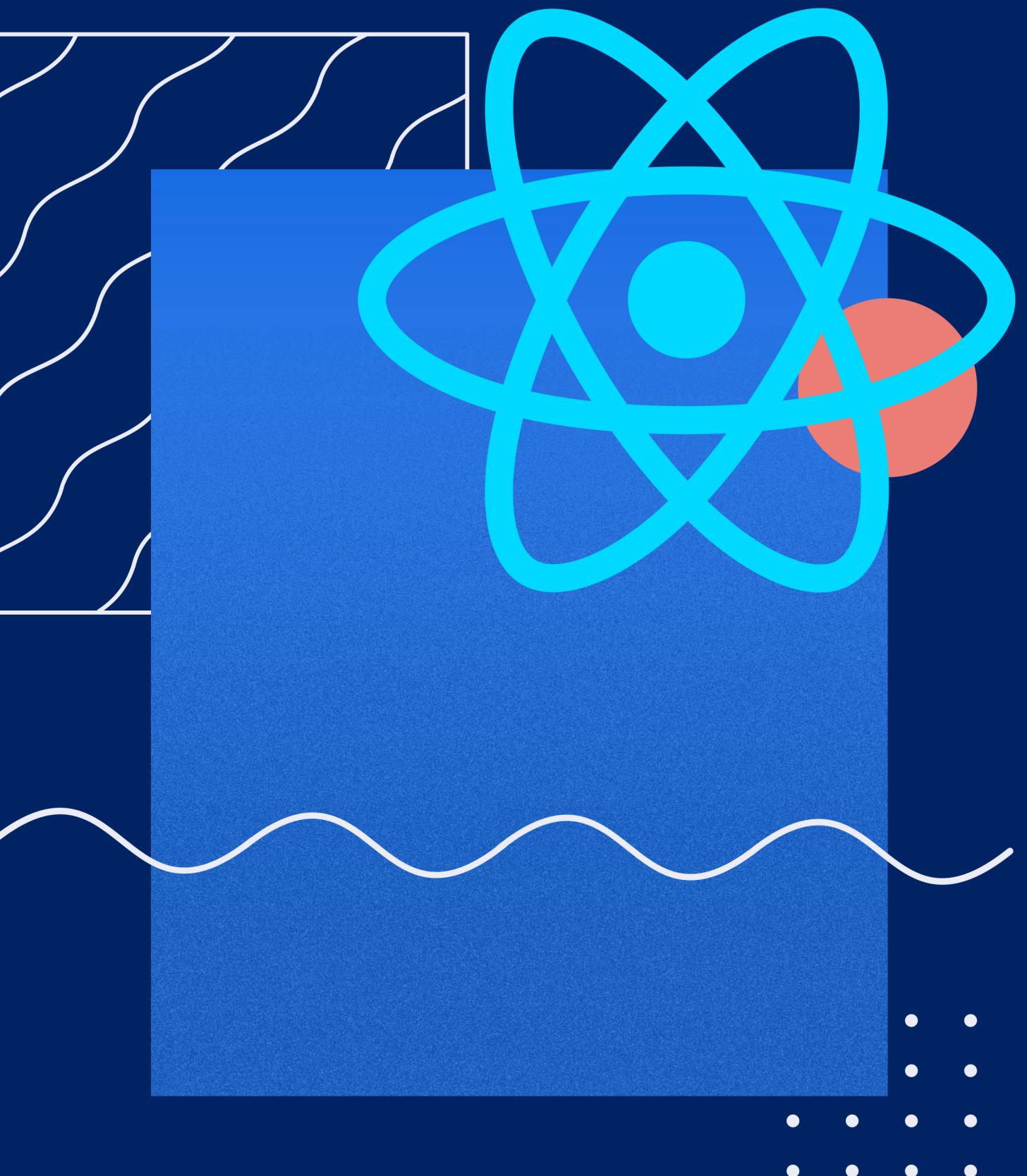
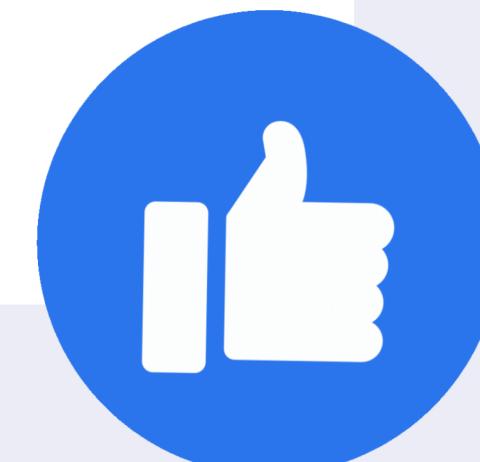
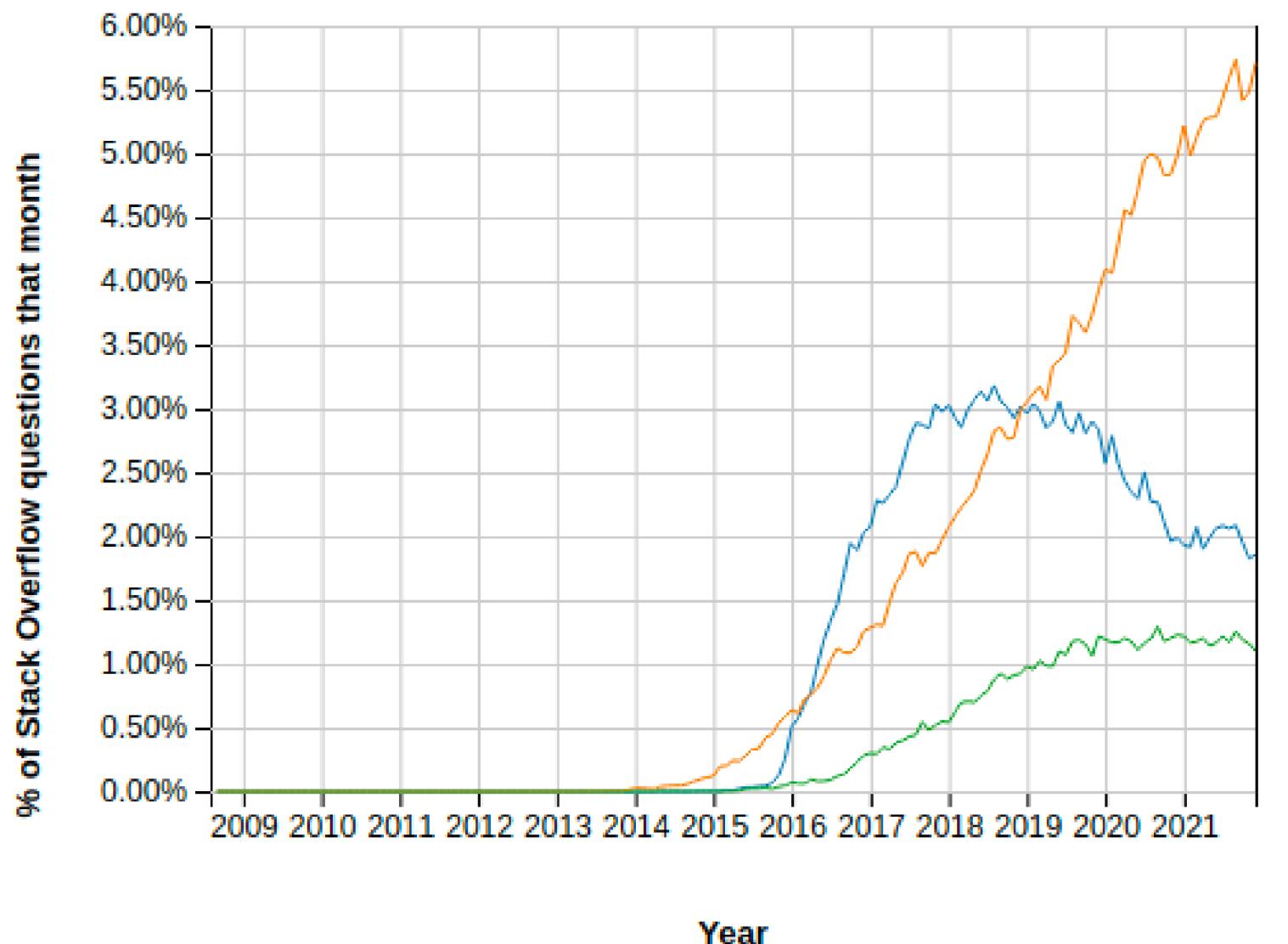


# Intro to React

DOCKER + CREATE-REACT-APP



# WHY REACT?



- Most popular front-end Javascript framework, released by Facebook in May 2013.
- Modular, component-based UI.
- Single page applications. Render changes to the browser through the Virtual DOM.

Image Source: **Stack Overflow Trends**  
(<https://insights.stackoverflow.com/trends?tags=angular%2Creactjs%2Cvue.js>)

# Old School Facebook (2004)

[ thefacebook ]

home search global social net invite faq logout

Scranton

Scott Pearson's Profile (This is you)

Picture [ edit ]



Information [ edit ]

**Account Info:**

Name: Scott Pearson, BS  
Member Since: January 12, 2005  
Last Update: February 3, 2005

**Basic Info:**

Email: persons2@scranton.edu  
Status: Alumnus/Alumna  
Sex: Male  
Year: 2004

Concentrations: Computing Sciences  
Mathematics

Phone: 570.499.4818  
High School: Dunmore HS '00

**Extended Info:**

Screenname: ScottiePP7  
Looking For: Friendship  
Dating  
A Relationship  
Random play  
Whatever I can get

Interested In: Women

Relationship Status: Single

Political Views: Liberal

Interests: Drinking, Football, Basketball, Tennis, saying you'll have that

quick search 00

My Profile [ edit ]  
My Groups  
My Friends  
My Messages  
My Away Message  
My Mobile Info  
My Account  
My Privacy

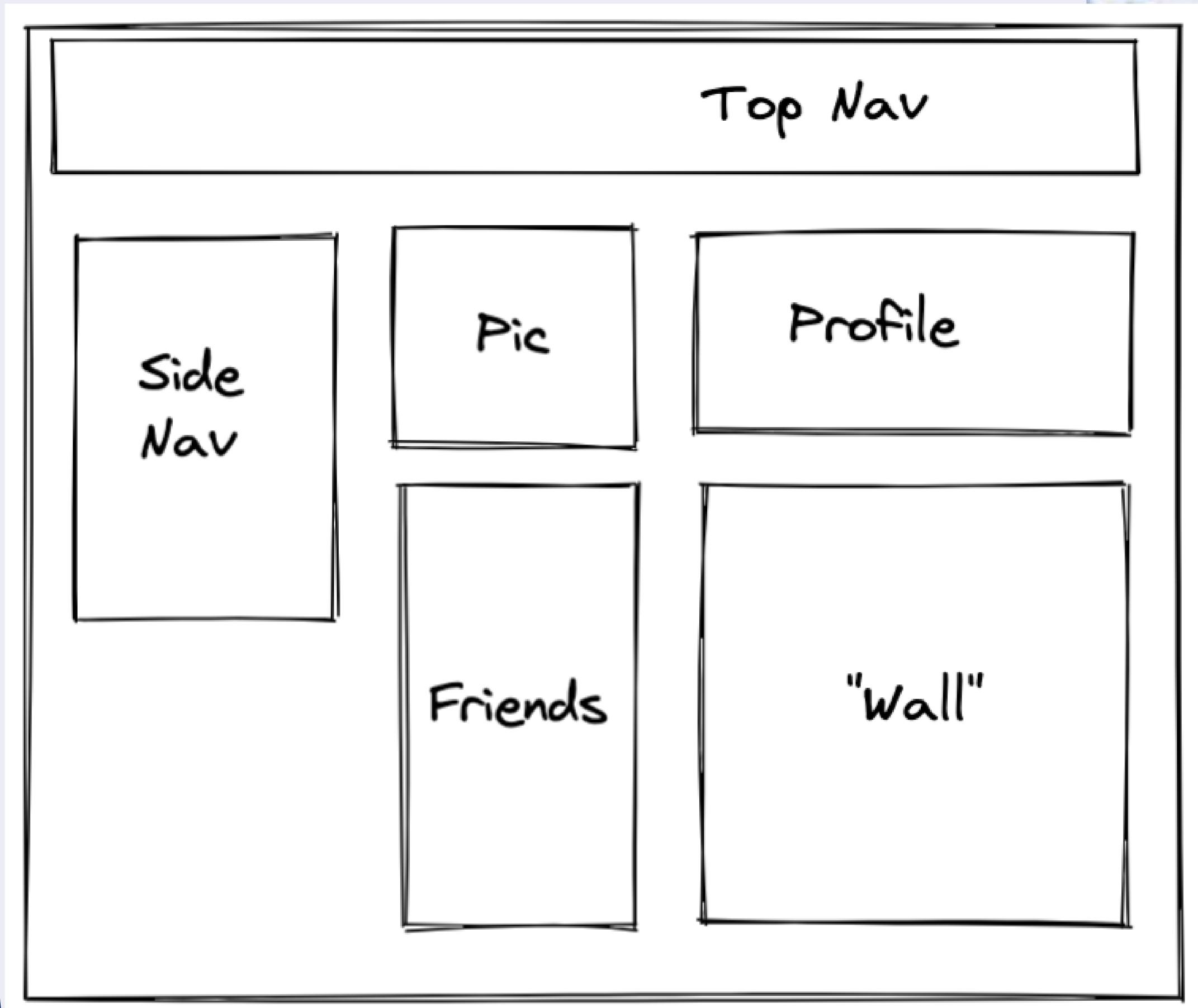
Visualize My Friends  
Edit My Profile  
My Account Preferences  
My Privacy Preferences

Connection  
This is you.

Access  
Scott is currently logged in from a non-residential location.

Other Scranton [ edit ]

# Old School Facebook



# Mini-Feed (née 2006)

facebook

Profile edit Friends Networks Inbox (8) home account privacy logout

Search Applications edit

- facehack
- Photos
- Groups
- Events
- Marketplace
- My Questions
- Top Friends
- + more

Gina Bianchini Profile

Update your status...

Networks: Stanford Alum

Mini-Feed edit See All

Displaying 10 stories.

Today

Gina added the Handbell Podcast Player application. 8:40pm

June 19

Gina and Paul Weller are now friends. 12:02pm

June 18

Gina and Jay Tannerbaum are now friends. 4:53pm

June 17

Gina added the Ning Network Creator's Video application. 7:52pm

Gina and Sam Jadallah are now friends. 7:51pm

Gina and Ian McCarthy are now friends. 7:51pm

Gina added the House of Kyle Latest Photos application. 7:50pm

Gina added the Top Friends application. 7:49pm

Gina added the My Questions application. 7:48pm

Gina and Brooke Hammerling are now friends. 7:47pm

Stanford Friends edit See All

13 friends at Stanford.

Valerie Rozycki Reid Hoffman David Sze

Steve Vassallo Ian McCarthy Hollie Moore

Information edit

Contact Info [edit]

Email: gina\_bianchini@stanfordalumni.org

To fill out the rest of your profile, click here.

Handbell Podcast Player edit X

Handbell Podcast 0711fs - Listener F

00:37 00:0000

Handbell Community

Share

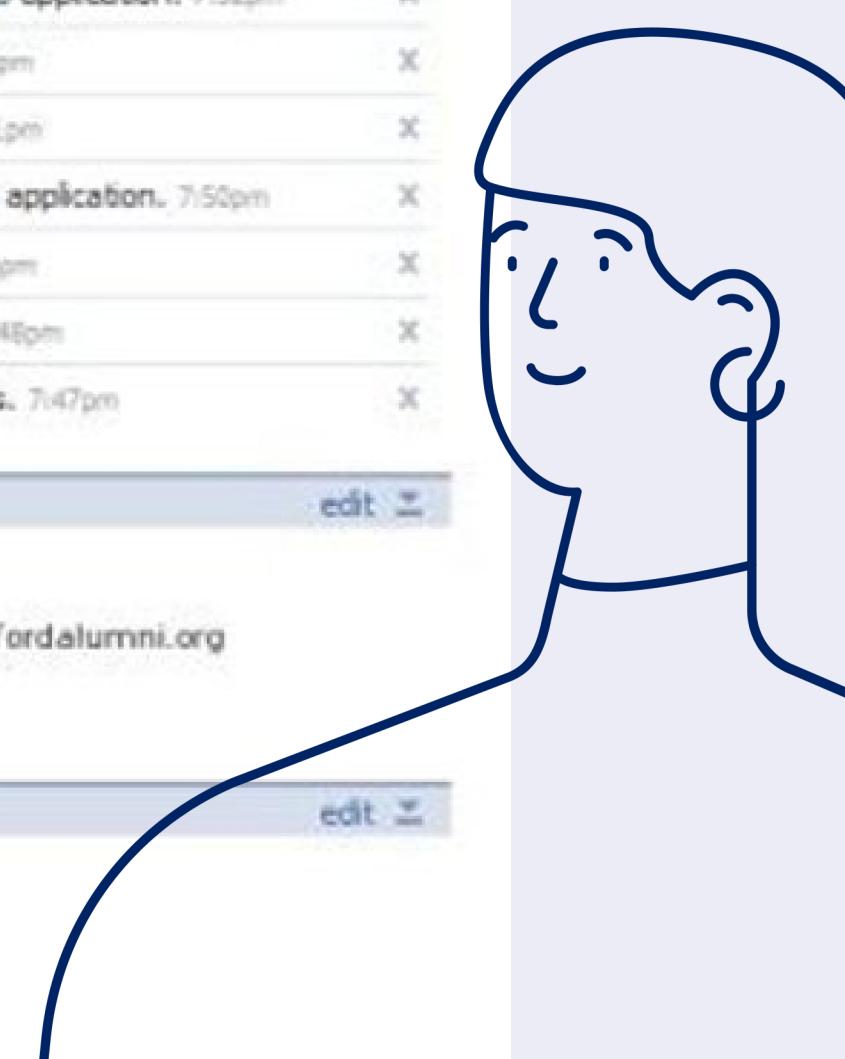
1. Handbell Podcast 0711fs - 2. Handbell Podcast 0711 - An

Education and Work edit

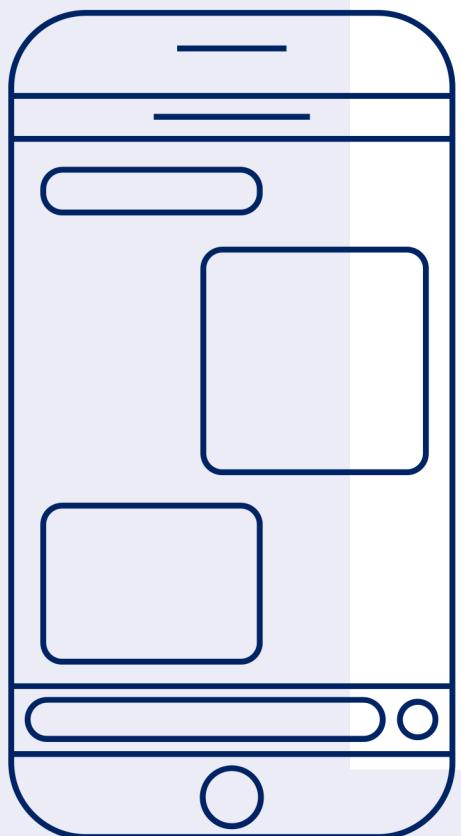
Education Info [edit]

College: Stanford Political Science

High School: Lynbrook High



# The News Feed



**facebook** Welcome Matt!

home search browse share invite help logout

Requests 1 group invitation

My Status edit

Keep your friends updated on your current status.

Shared with You see all

ShowBizSpy - Movies From Eddie Lim

Microsoft launches the Zune! - Engadget From Soleio

Drew From Soleio

Birthdays see all

Today's Birthdays Kyle

October 7th Birthdays Joe Alisha Nick

October 8th Birthdays

News Feed

Sponsored: iTunes

Athalie and Meredith commented on the note tab energy: an anomaly in my life. 3:59pm

Carrie joined the group Marvel: Ultimate Alliance – The World's Largest Army of Super Heroes. 3:39pm

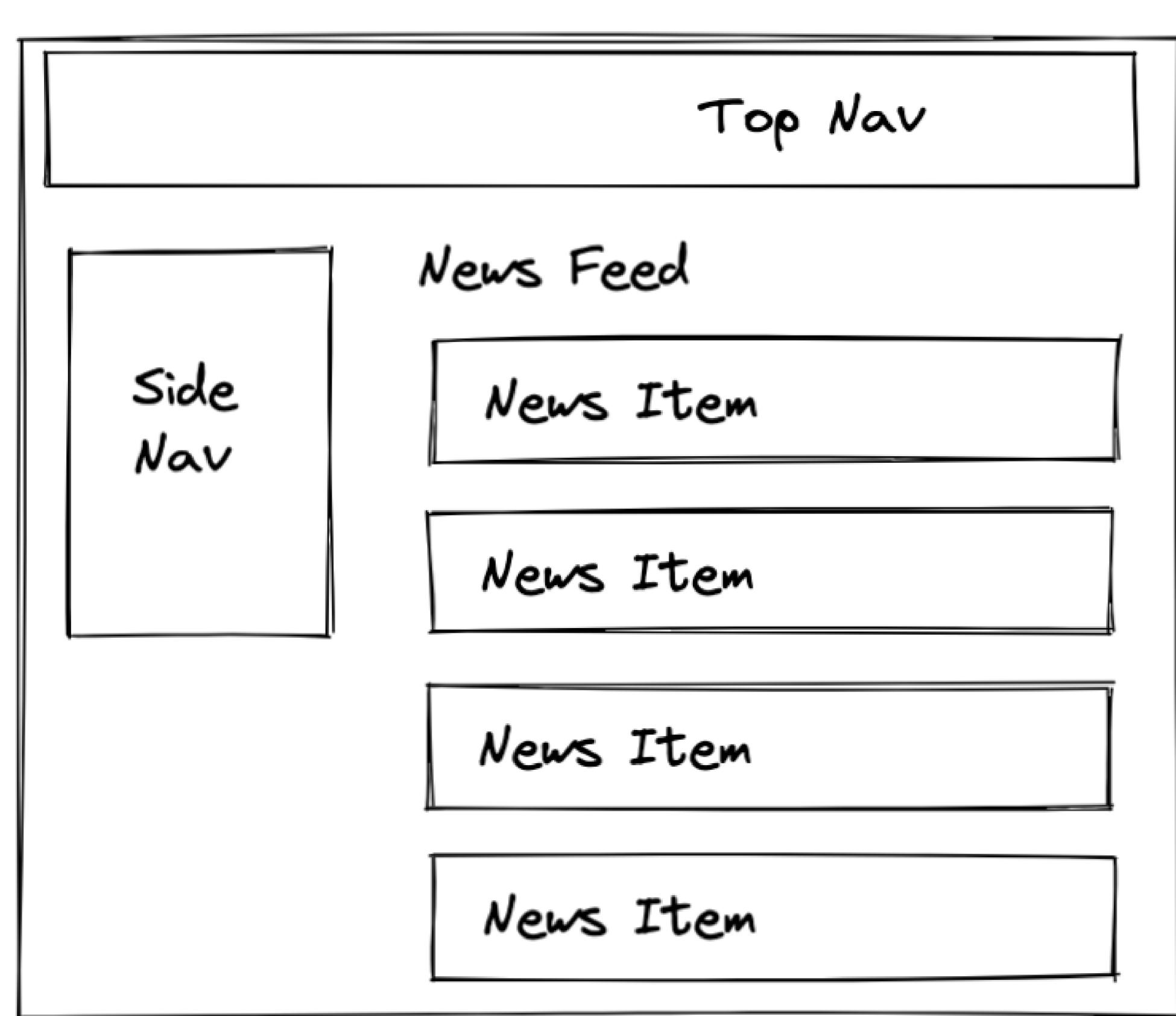
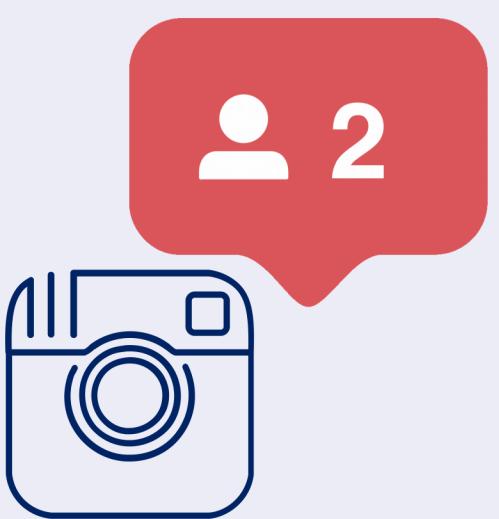
Payam Imani added new photos. 3:30pm Share Save

Gotham City: Chicago – 58 photos by Payam Location: Chicago, IL In this album: Payam

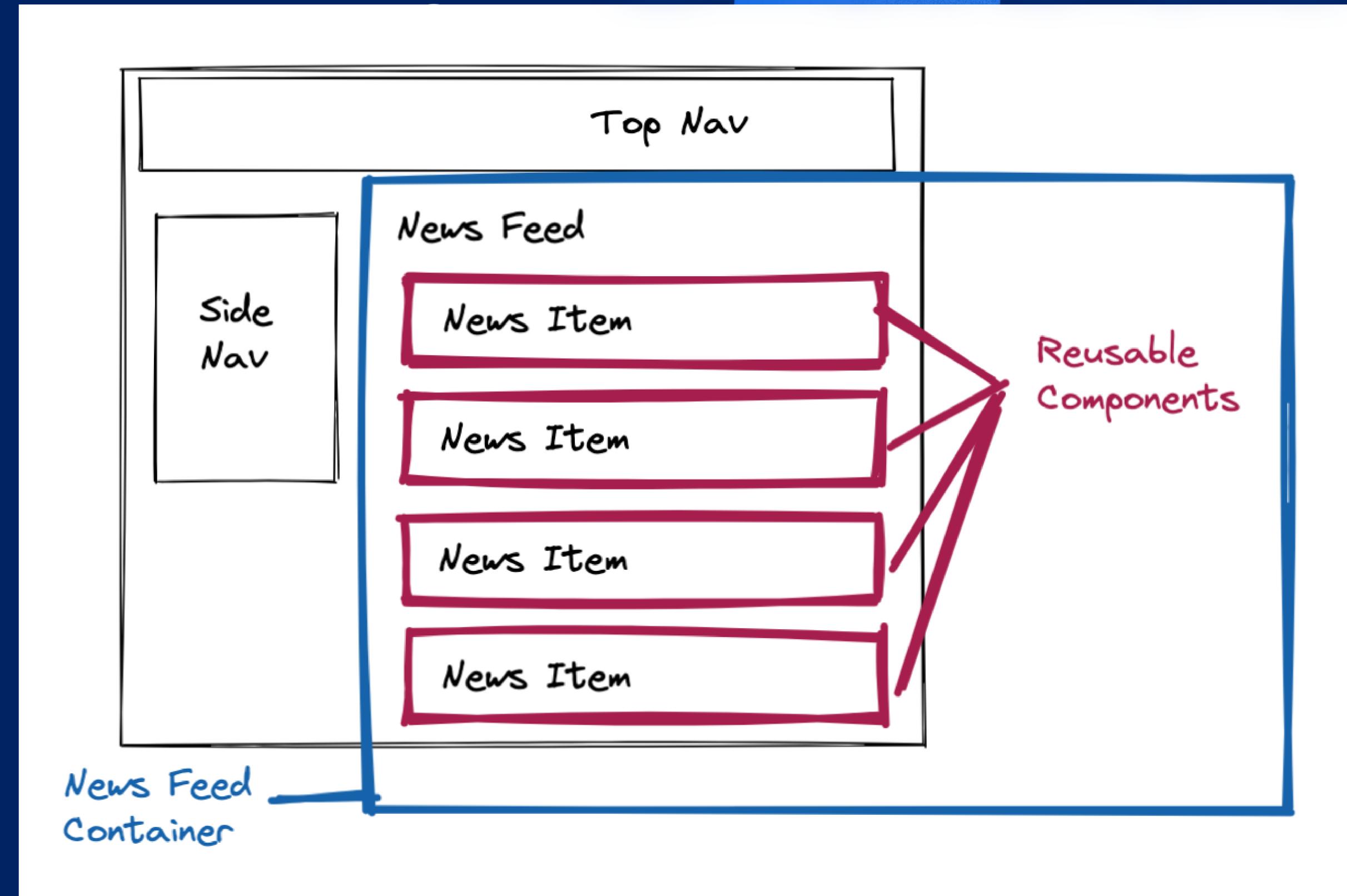
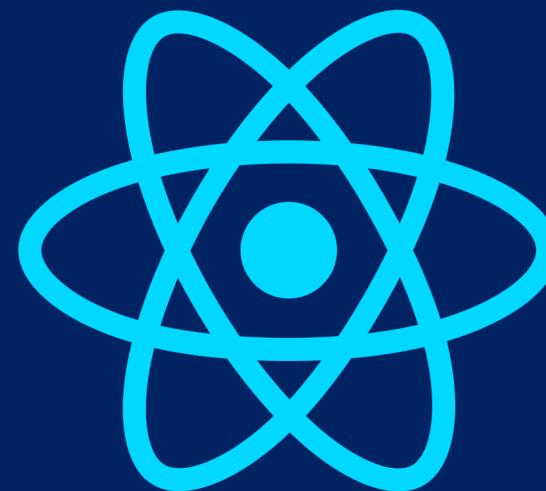
Steve created a group. 2:59pm

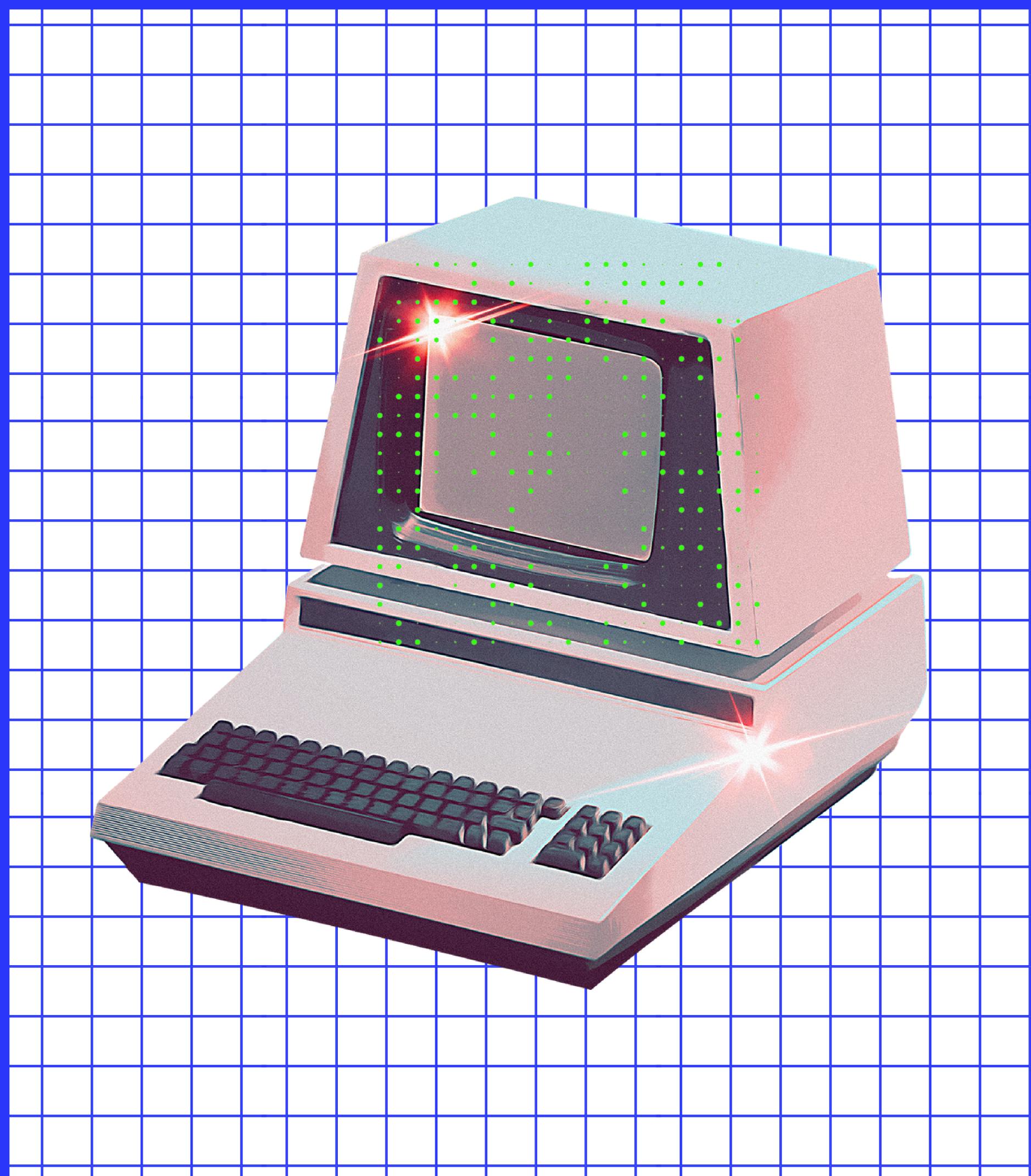
# Infinite Scroll

React.js created by  
Facebook engineer  
Jordan Walke (2011)



# React Components





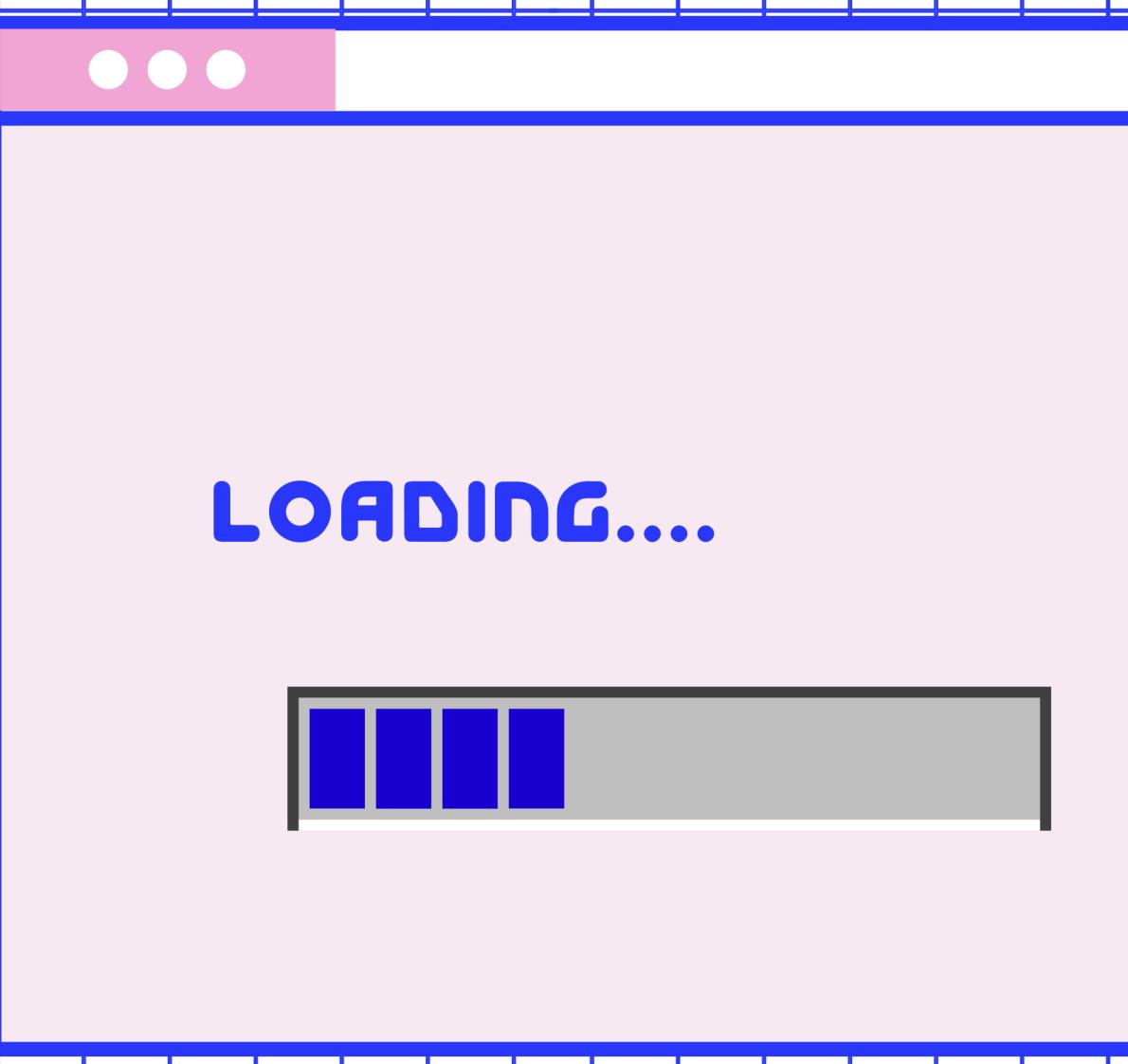
OK COMPUTER!

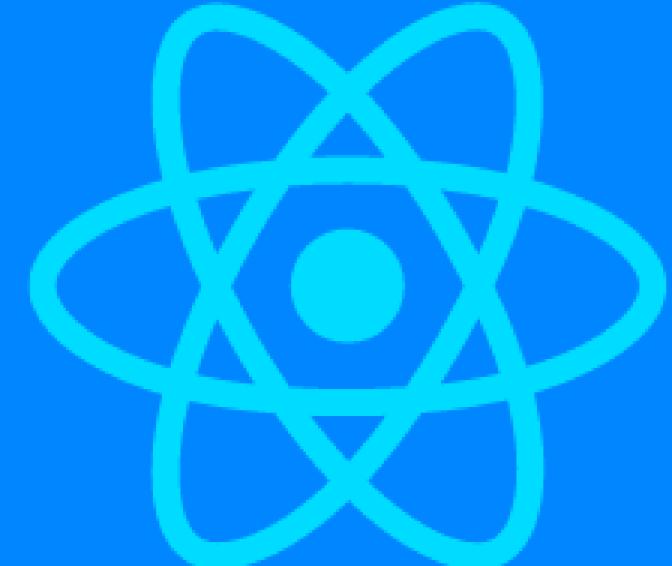
DEMO

Building a Dockerfile for using React

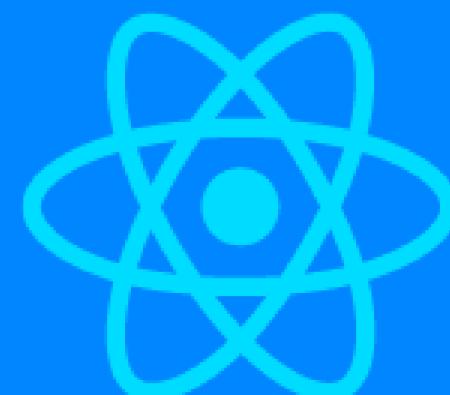
CREATE-REACT-APP

A presentation slide is shown. At the top left is a red bar with three white dots. At the top right is the text "OK COMPUTER!". In the center is a large blue button with the word "DEMO" in white. Below the button is the text "Building a Dockerfile for using React". At the bottom right is the text "CREATE-REACT-APP". The slide has a white background with a thin black border.

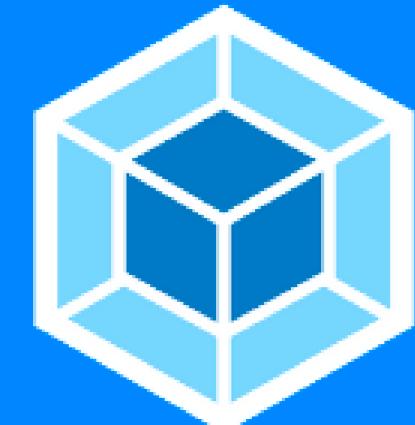




# Create React App



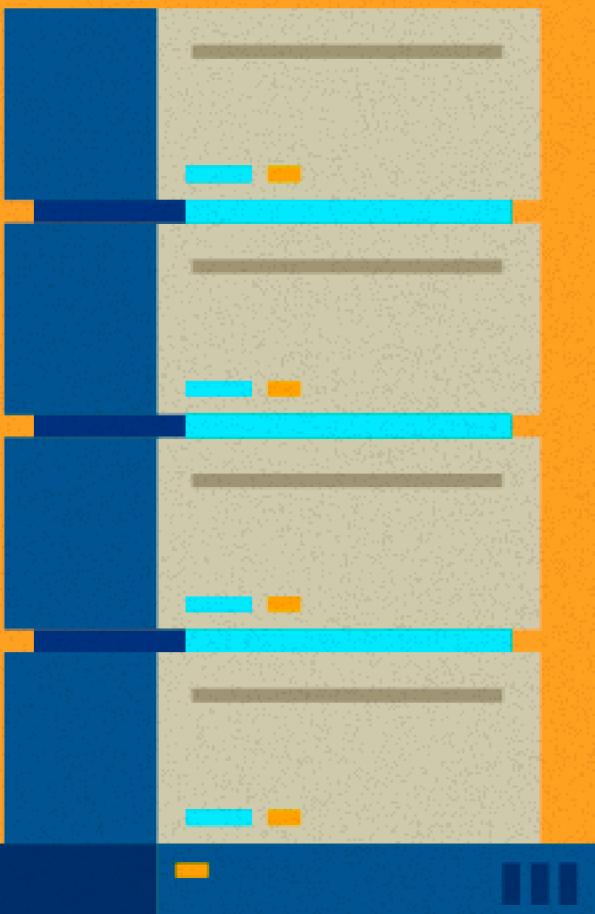
**BABEL**



**webpack**

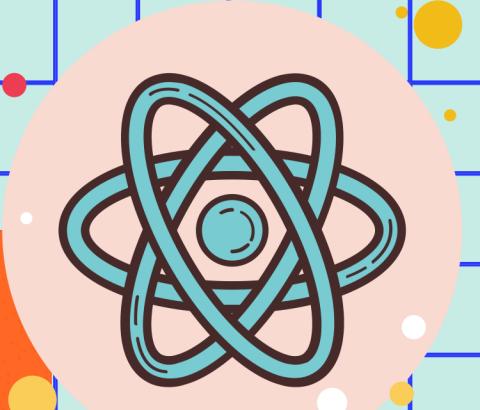
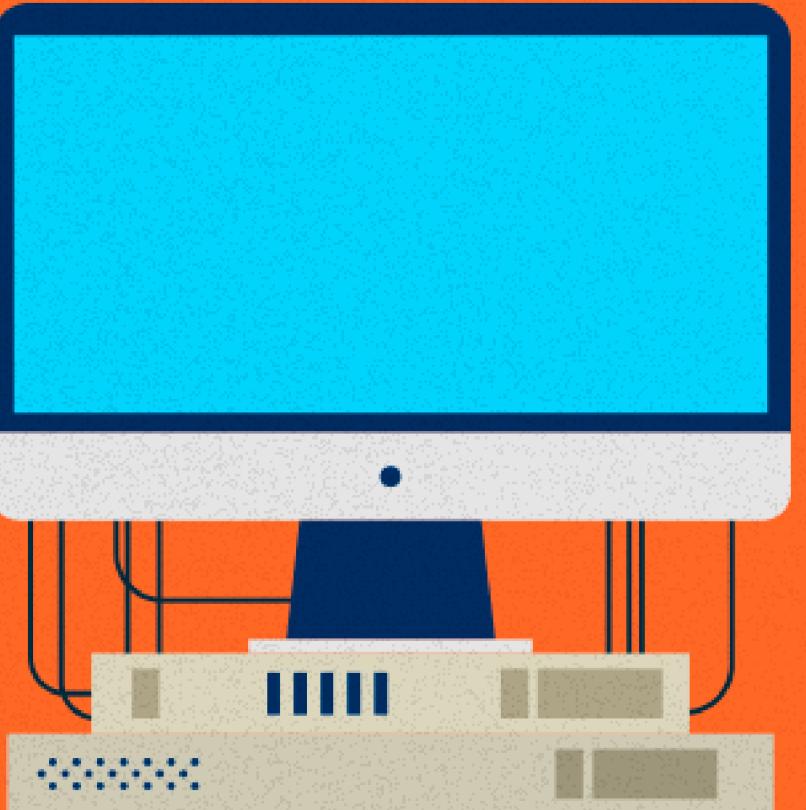
dj

SERVER  
SIDE  
RENDERING



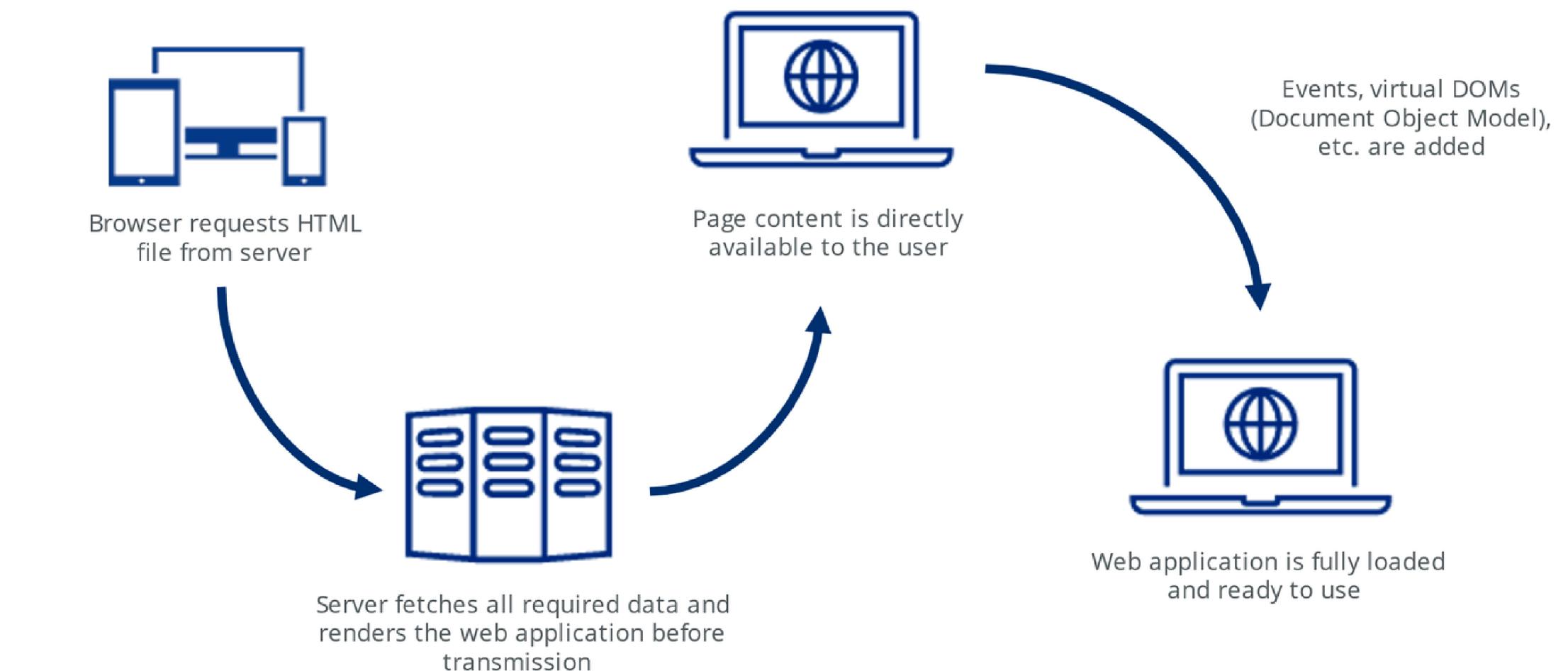
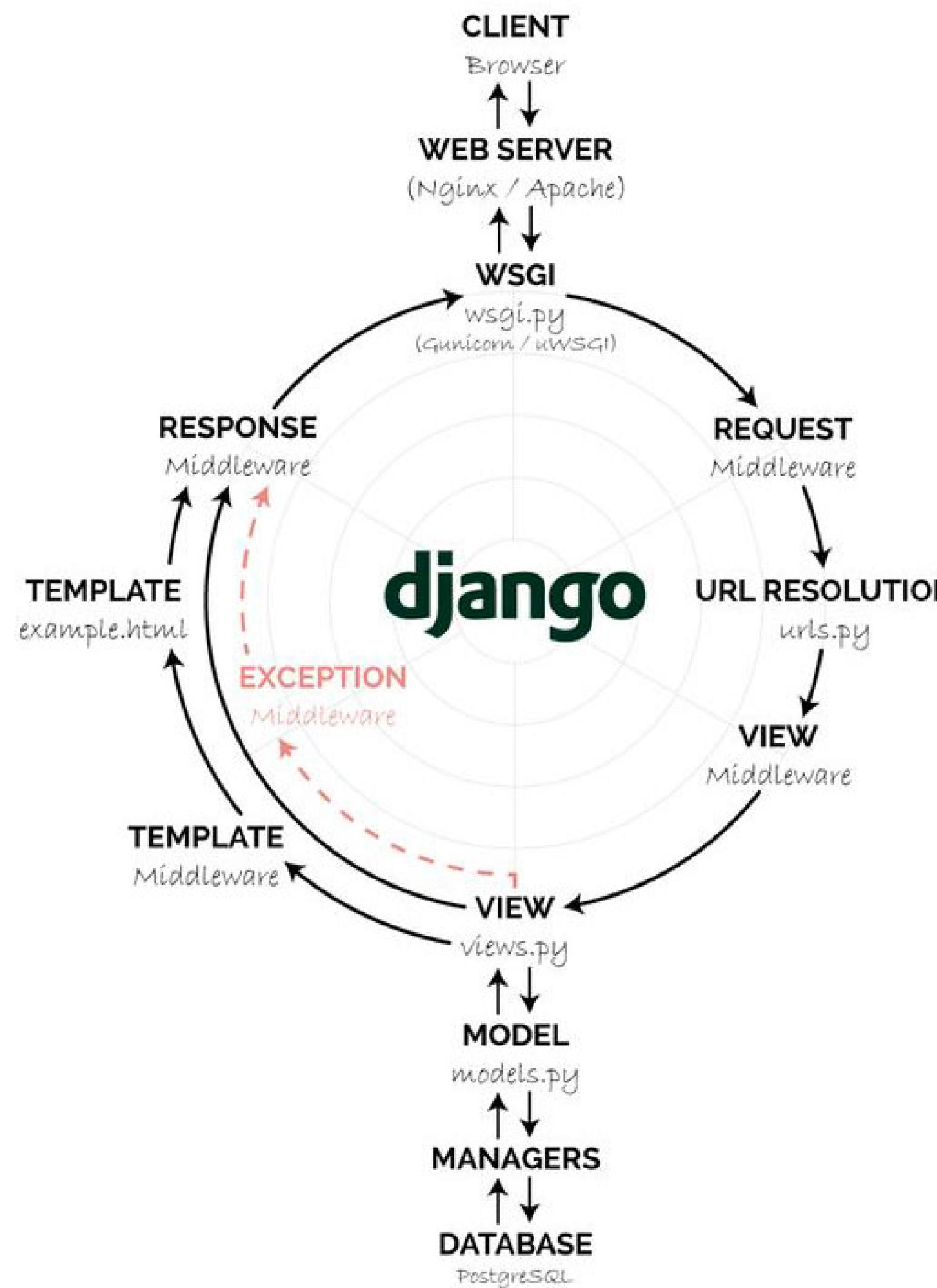
VS

CLIENT  
SIDE  
RENDERING

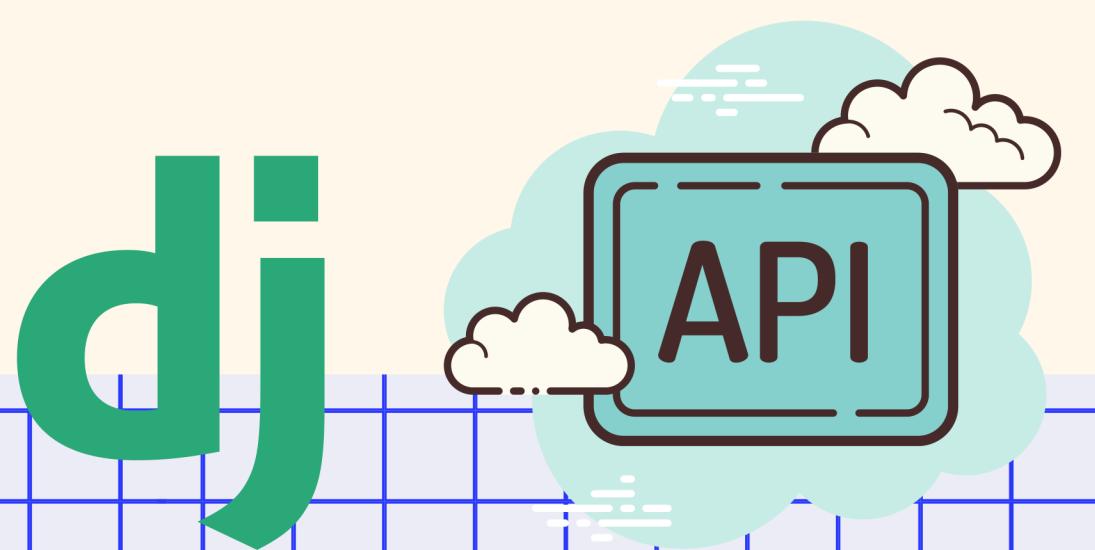
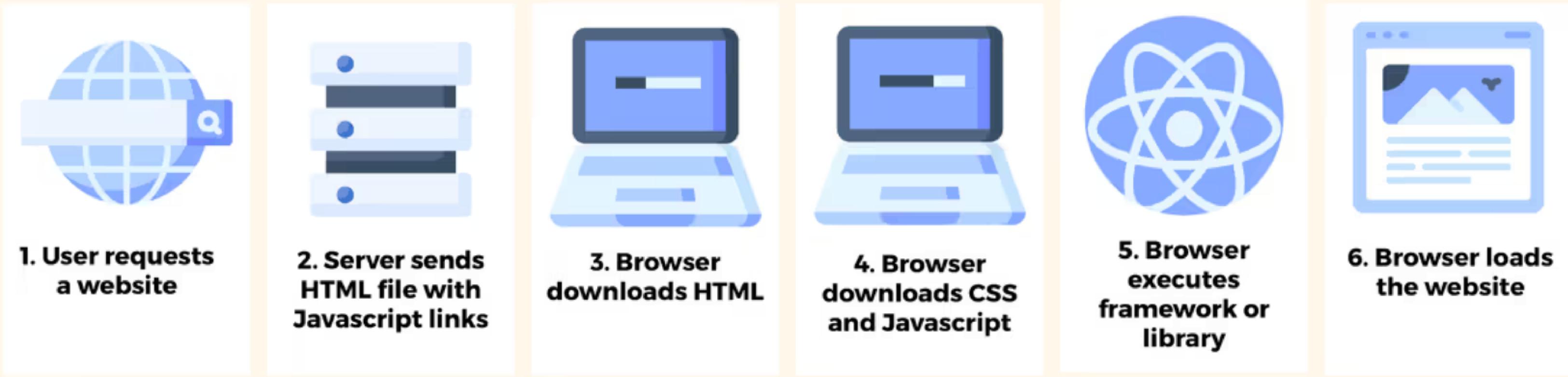
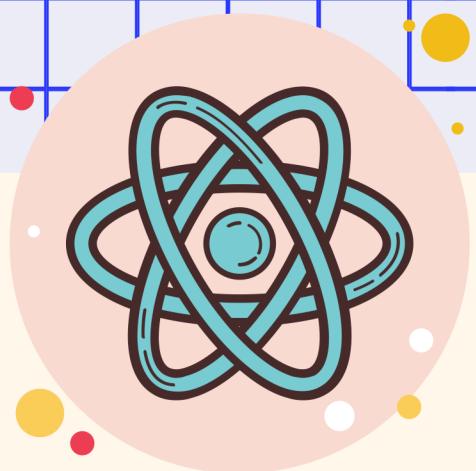




## Server side rendering



# Client Side Rendering



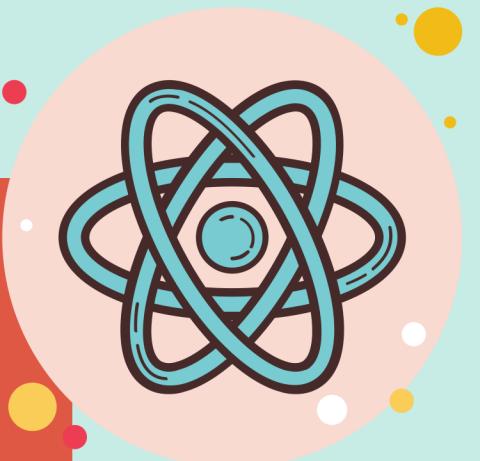


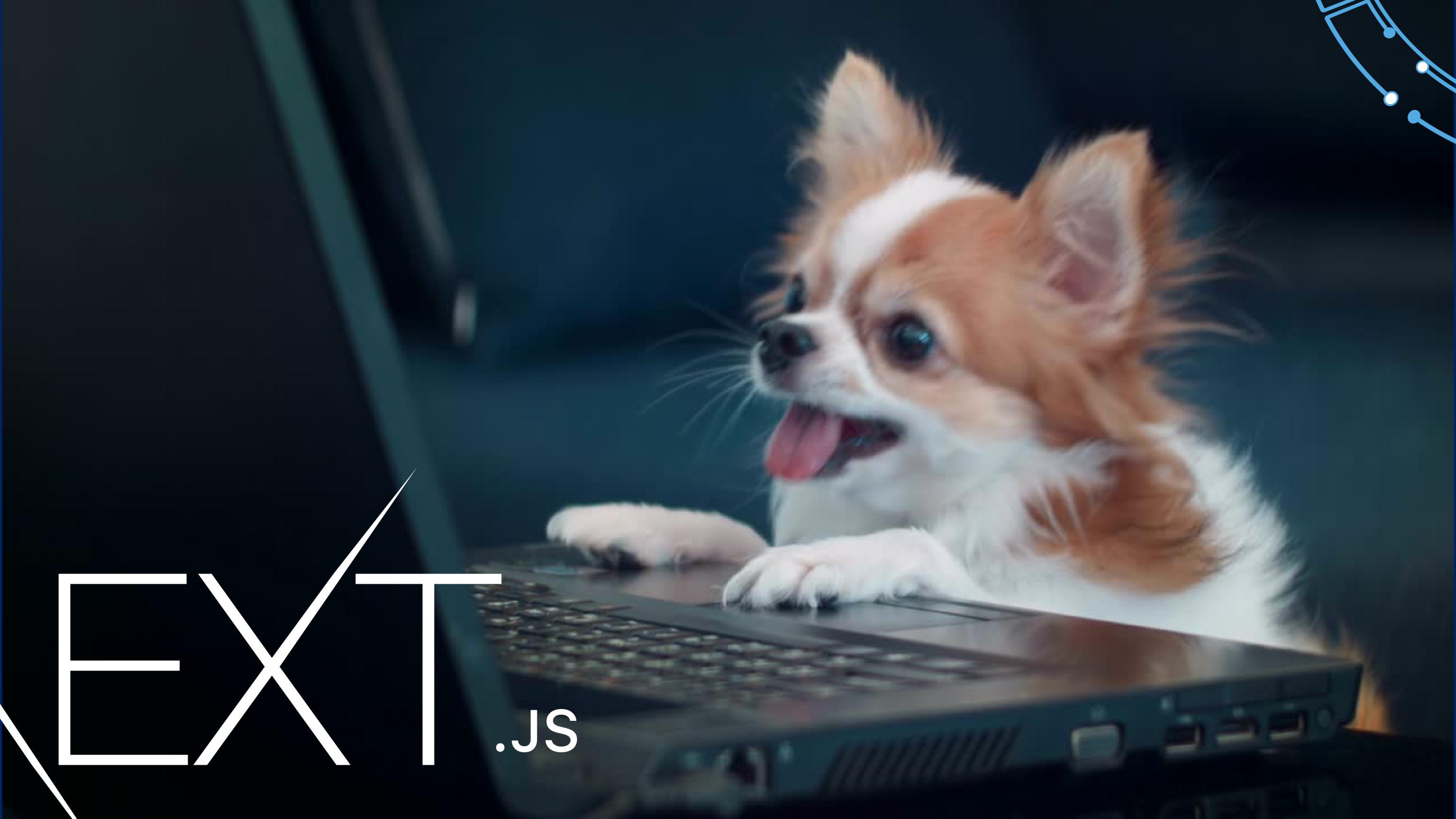
## SSR

- + Ideal for static sites
- + Quick initial page access
- + No JS dependency
- + Easy-to-crawl site for better SEO
- Multiple server requests
- Full page reloads
- Non-rich site interactions
- Higher latency, prone to vulnerability

## CSR

- + Ideal for web apps
- + Fast site rendering after initial load
- + Rich site interactions
- + Robust JS library selection
- Negative SEO for incorrect rendering & API response delays
- Slow initial load time
- Requires external library
- Higher memory consumption





NE  
XT  
.JS



OK COMPUTER!

# FUNCTIONAL COMPONENTS

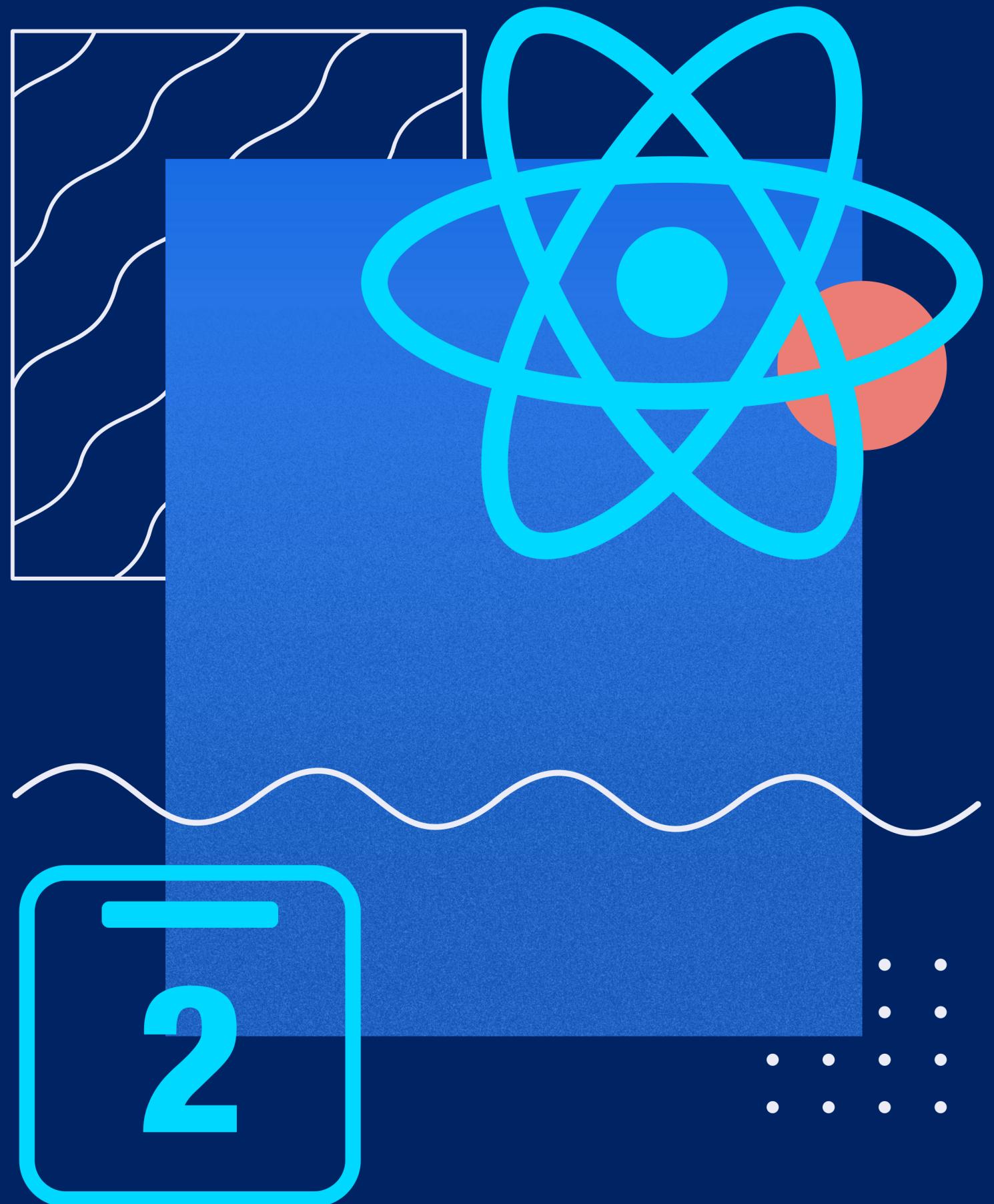
Build a newsfeed.

CREATE-REACT-APP

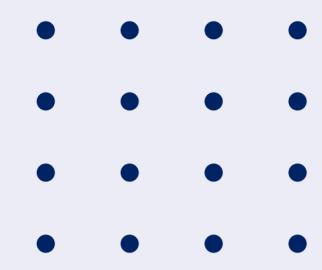


# Stateful Components

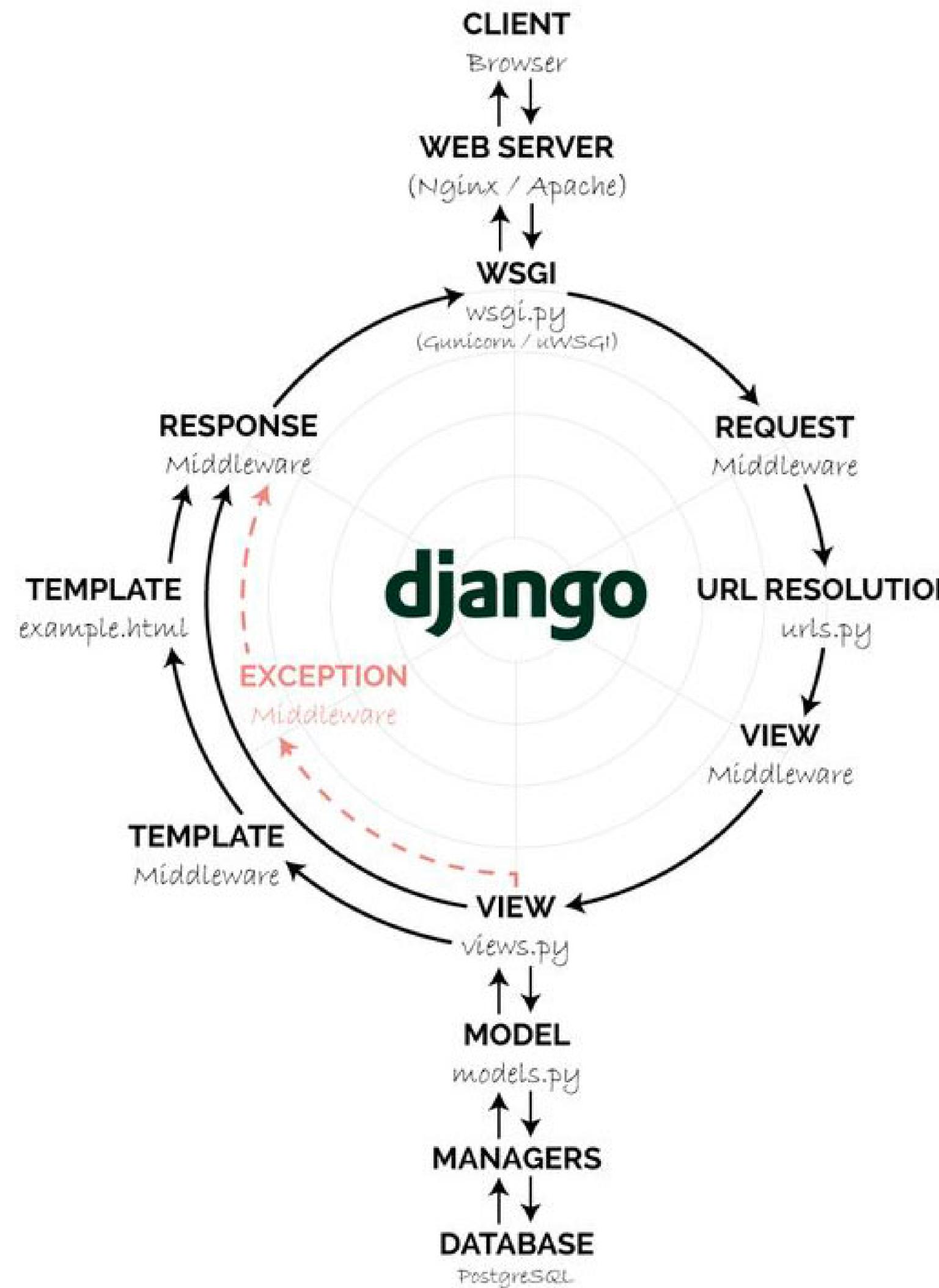
BUILDING A FORM IN REACT.



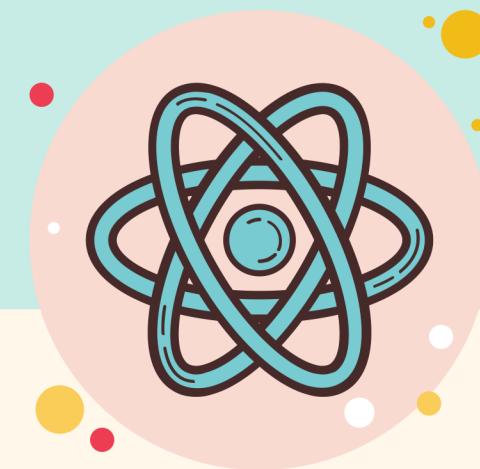
# REVIEW



- What is React: front-end framework for creating modular, component-based UI.
- Docker + React
- How React integrates with Django
- Writing functional components



# Front End



## Client Side Rendering



# Back End

# Dockerfile

```
# Use the latest version of Node.js
FROM node:latest

# Use the /app directory as the work directory
WORKDIR /app

# Install the tool to create React apps
RUN npm install create-react-app
```

# Terminal

```
docker build . -t react-docker
```

```
docker run -it -v "$(pwd):/app" -p 3000:3000 react-docker bash
```

```
npx create-react-app example  
cd example  
npm start
```

# create-react-app

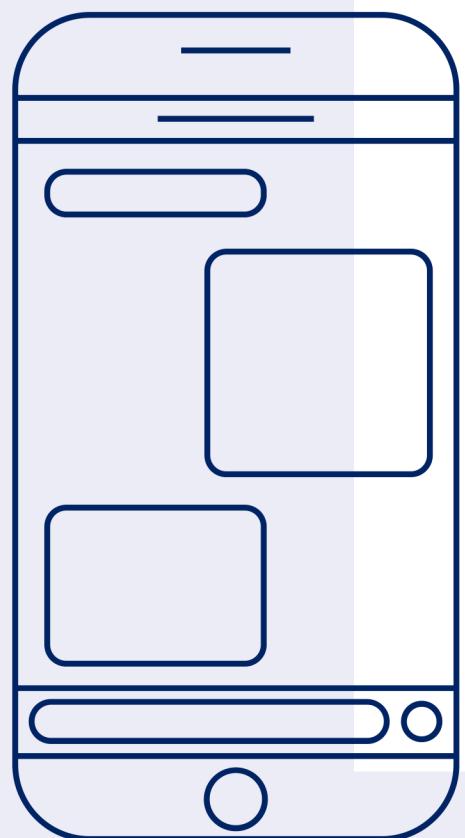
npm run build

npm run start

A screenshot of a file explorer window titled "DEMO". The tree view shows the following structure:

- > build
- ✓ public
  - ★ favicon.ico
  - ↳ index.html
  - 🖼 logo192.png
  - 🖼 logo512.png
  - { manifest.json
  - ≡ robots.txt
- ✓ src
  - # App.css
  - JS App.js**
  - JS App.test.js
  - # index.css
  - JS index.js
  - ─ logo.svg
  - JS setupTests.js
  - ↳ .gitignore
- { package-lock.json
- { package.json
- ⓘ README.md

# The News Feed



**facebook** Welcome Matt!

home search browse share invite help logout

**News Feed**

Sponsored: iTunes

Athalie [Lapierre](#) and Meredith [Cline](#) commented on the note tab energy: an anomoly in my life. 3:59pm

Carrie [Hawkins](#) joined the group Marvel: Ultimate Alliance – The World's Largest Army of Super Heroes. 3:39pm

Payam Imani added new photos. 3:30pm [Share](#) [Save](#)

Gotham City: Chicago – 58 photos by Payam [Imani](#) Location: Chicago, IL In this album: Payam

Steve [Wilkerson](#) created a group. 2:59pm

Requests 1 group invitation

My Status [edit](#) Keep your friends updated on your current status.

Shared with You [see all](#)

ShowBizSpy - Movies From Eddie Lim

Microsoft launches the Zune! - Engadget From Soleio

Drew [Hawkins](#) From Soleio

Birthdays [see all](#)

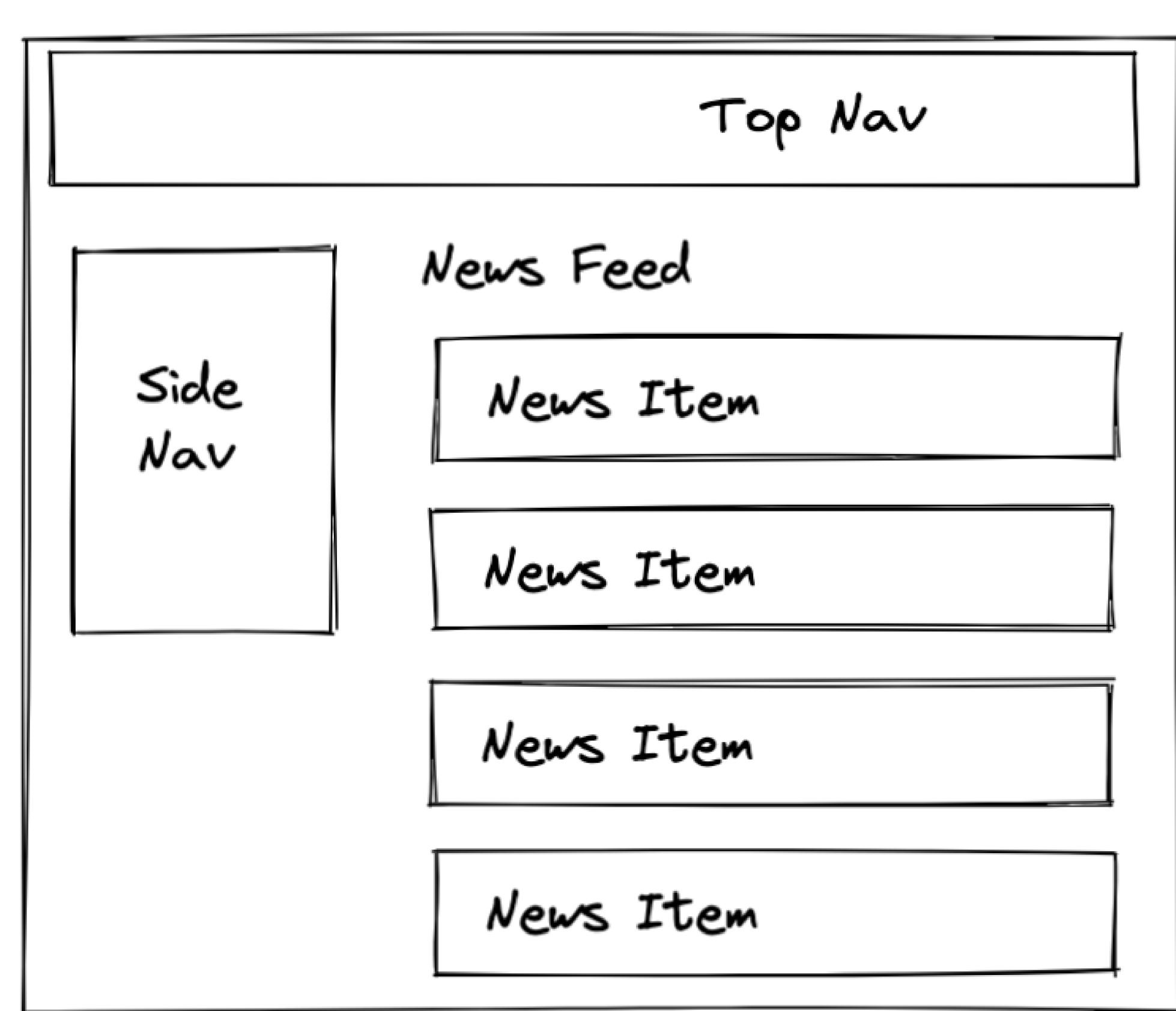
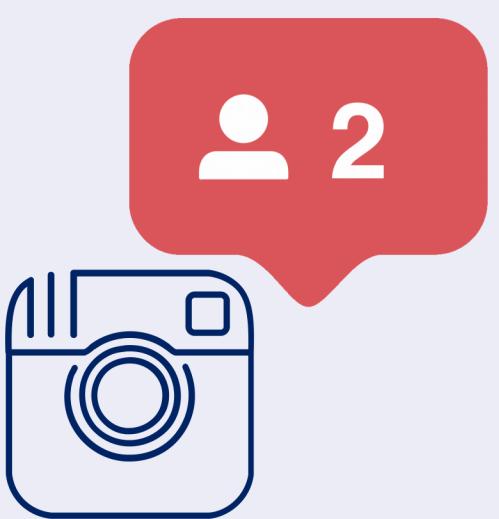
Today's Birthdays Kyle [Hawkins](#)

October 7th Birthdays Joe [Wilkerson](#) Alisha [Kroder](#) Nick [Hawkins](#)

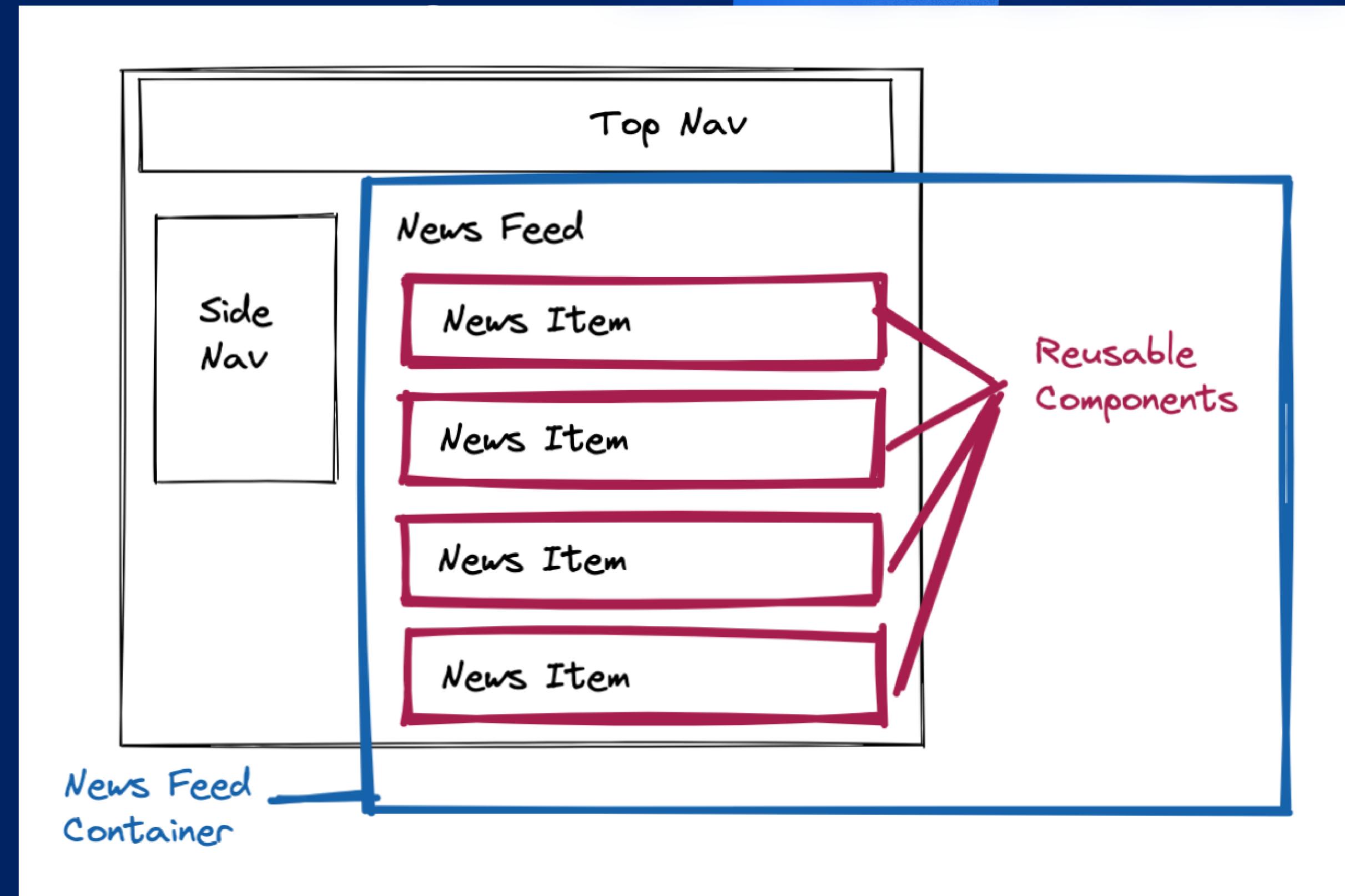
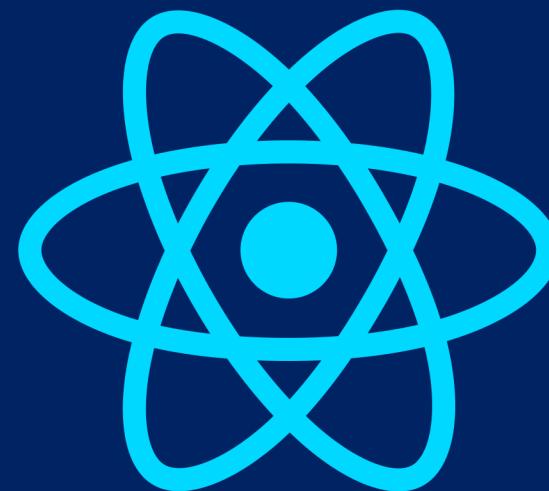
October 8th Birthdays

# Infinite Scroll

React.js created by  
Facebook engineer  
Jordan Walke (2011)



# React Components



# Component Hierarchy

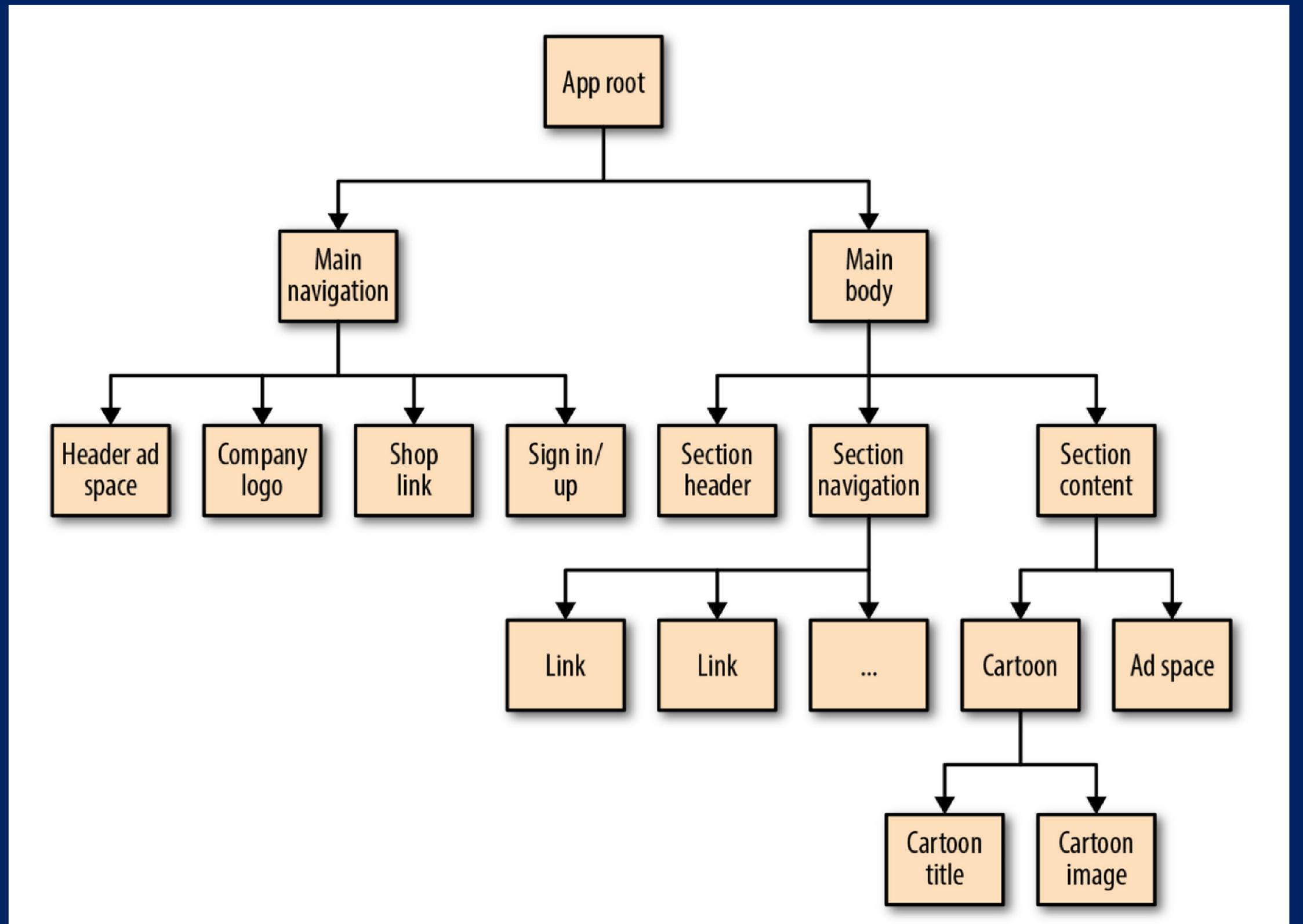


Image Source: **O'Reilly**  
<https://www.oreilly.com/library/view/what-react-is/9781491996744/ch01.html>



# Passing Props

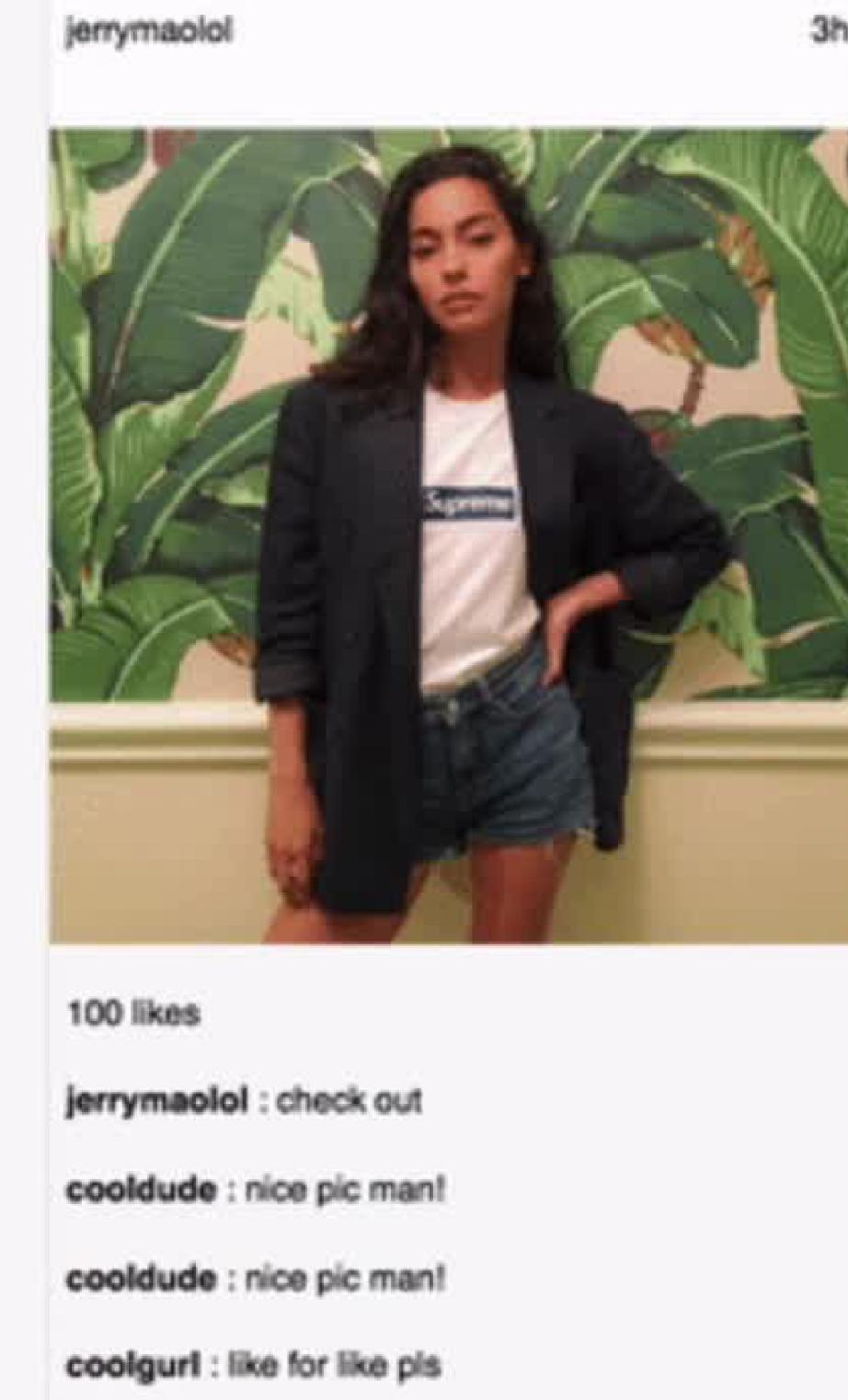
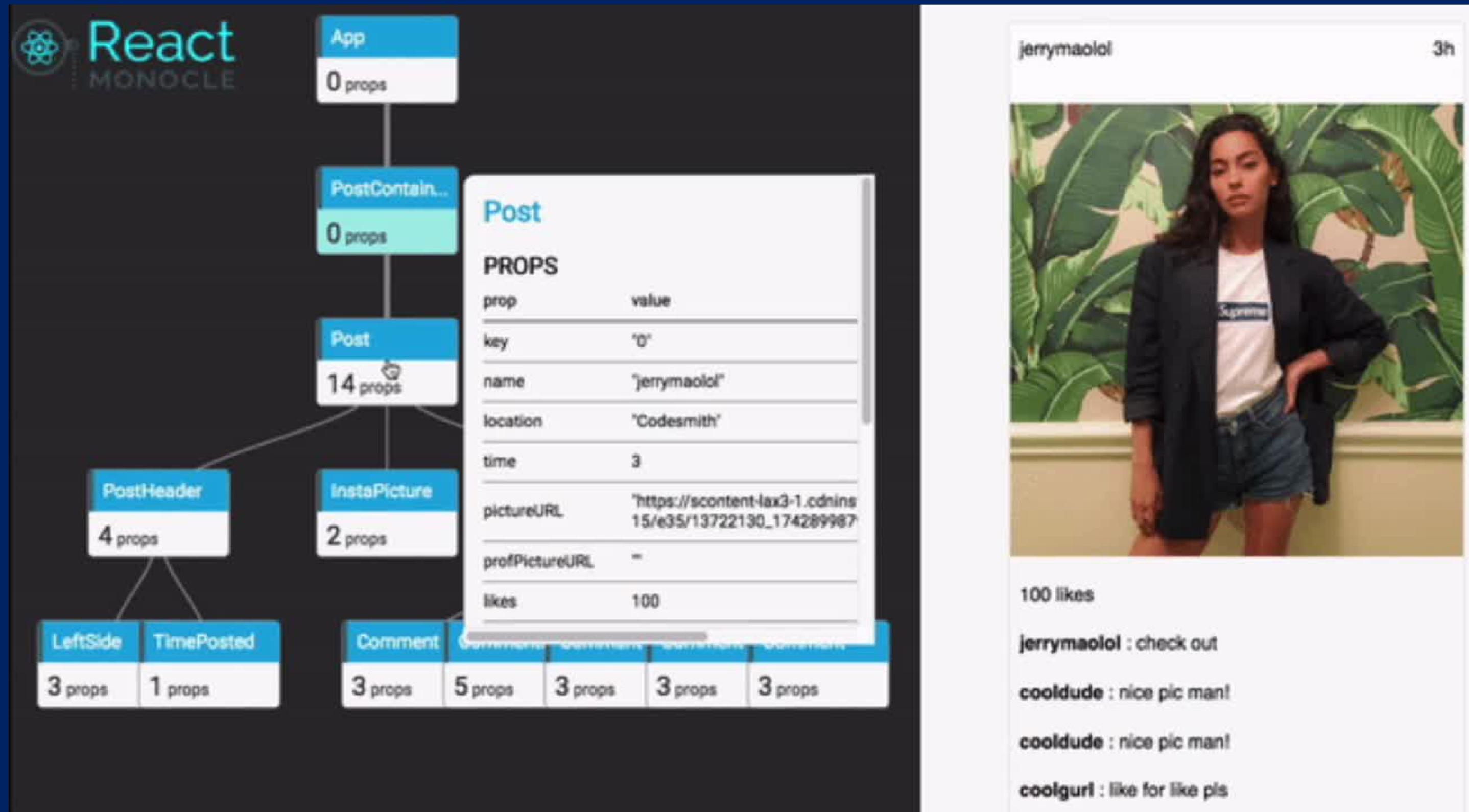


Image Source: [React Monocle](https://github.com/team-gryff/react-monocle)  
(<https://github.com/team-gryff/react-monocle>)

# PROPS

Props are data passed from a parent component to a child component.

Props are passed via HTML attributes and are used in React components as arguments.

Props are immutable.

# Passing Props

Parent component (NewsFeed.jsx)

```
<div className='NewsFeed'>
  {data.map((item, index) => {
    return(
      <NewsItem key={index} name={item['name']} action={item['action']} />
    )
  )}
</div>
```

# Passing Props

Child component (NewsItem.jsx)

```
1  function NewsItem(props) {
2      const {name, action} = props; <----- Javascript Object Destructuring
3      return(
4          <div className='NewsItem'>
5              <h3>{name}</h3>
6              <p>{action}</p>
7          </div>
8      );
9  }
10
11 export default NewsItem;
```

# FUNCTIONAL COMPONENT

```
src > components > example.jsx > ...
1 import React from 'react'
2
3 export default function example() {
4     return (
5         <div>
6             <h1>Hello World</h1>
7         </div>
8     )
9 }
10
```



# CLASS-BASED COMPONENT

```
src > components > ⚙ example.jsx > 🏃 example > 📁 render  
1 import React, { Component } from 'react'  
2  
3 export default class example extends Component {  
4     render() {  
5         return (  
6             <div>  
7                 <h1>Hello World</h1>  
8             </div>  
9         )  
10    }  
11 }  
12
```



# STATE

The `state` object is a mutable store of data inside the component.

To change a value in the state object, use the `this.setState()` method.

When a value in the state object changes, the component will re-render.

# Setting State

```
class HelloWorld extends React.Component {  
  constructor(props) {  
    super(props);  
    this.state = {  
      name: 'Woof! 🐶'  
    };  
  }  
  render() {  
    return (  
      <>  
        <h1>Hello, {this.state.name}</h1>  
        <input onChange={event => this.setState({name: event.currentTarget.value})}/>  
      </>  
    );  
  }  
  export default HelloWorld;
```

# CLASS / FUNCTIONAL

```
import React from "react";

class ClassHelloWorld extends React.Component {
  render() {
    return <h1>Hello, World</h1>;
  }
}

const FunctionHelloWorld = function () {
  return <h1>Hello, World</h1>;
};

const ArrowHelloWorld = () => <h1>Hello, World</h1>;
```

# Lifecycle Methods

A component “mounts” when it renders for the first time.

```
componentDidMount() {  
  fetchPosts().then(response => {  
    this.setState({  
      posts: response.posts  
    });  
  });  
  
  fetchComments().then(response => {  
    this.setState({  
      comments: response.comments  
    });  
  });  
}
```

**componentDidMount()**  
**componentWillMount()**  
**componentDidUpdate()**  
**componentWillUnmount()**

# Event Handling

.onClick()  
.onSubmit()

```
<div>
  <button onClick={this.showAlert}>Click Me</button>
</div>
```

Event **listener** names are camelCase instead of lowercase, and we pass a function in as the event handler, rather than a string.

```
showAlert(e) {
  e.preventDefault();
  alert("hello");
}
```

# Binding "This"

```
this.handleChange = this.handleChange.bind(this);  
this.handleSubmit = this.handleSubmit.bind(this);
```

The default binding for "this" is the browser window. In order to avoid this possible problem at runtime, we explicitly bind the method to the object.

**READ MORE HERE:**

<https://www.freecodecamp.org/news/this-is-why-we-need-to-bind-event-handlers-in-class-components-in-react-f7ea1a6f93eb/>