

Web fundamentals for performant web apps

JSDC.TW
December 2024

Dhrubajit Paul
Chowdhury
@dhrubajit
Ascenda



← Post



Kenneth Cassel

@KennethCassel

...

how come a company founded over 100 years ago has the fastest site on the internet?

The screenshot shows the McMaster-Carr website interface. On the left, a sidebar lists categories such as Abrading & Polishing, Building & Circuits, Electrical & Lighting, Fabricating, Fastening & Joining, Filtering, Flow & Level Control, Furniture & Storage, Hand Tools, Hardware, Heating & Cooling, Lubricating, Material Handling, Measuring & Inspecting, Office Supplies & Signs, Pipe, Tubing, Hose & Fittings, Plumbing & Joints, Power Transmission, Pressure & Temperature Control, Pulling & Lifting, Raw Materials, Safety Supplies, Saws & Cutting, Sealing, Shipping, and Suspending. The main content area displays several product categories with corresponding icons and labels:

- Fastening & Joining:** Screws & Bolts, Threaded Rods & Stacks, Grommets, U-Bolts, Nuts, Washers, Shims, Helical & Threaded Inserts, Spacers & Standoffs, Pins, Anchors.
- Adhesives & Tape:** Adhesives, Tape, Hook & Loop.
- Welding, Braizing & Soldering:** Electrodes & Wires, Welders, Gas Regulators, Welding Gloves, Welding Helmets & Glasses, Protective Screens, Braising Alloys, Ties, Beldor, Belding Irons, Melting Pots.
- Pipe, Tubing, Hose & Fittings:** Pipe Fittings & Flex, Pipe Hangers, Pipe Joints, Pipe & Tube Repair Clamps, Pipe Flange Spreaders, Tubing, Tube Fittings, Tube Cutters, Tube Bending Tools, Tube Benders, Manifolds, Hose, Hose Fittings, Hose Hoses, Hose & Tube Clamps, Hose Reels, Tank Fittings.

A timer at the bottom left indicates the video is at 0:22. The top right corner shows the user's name, Kenneth Cassel, and other account details.

→ x.com/KennethCassel/status/1847034096062710087

→ www.mcmaster.com

"Performance Doesn't Directly Affect Revenue"

"The Website is Fast Enough Already"

"It's Just for a Small Percentage of Users"

"We Have Bigger Priorities"

"It's Not a User Priority"



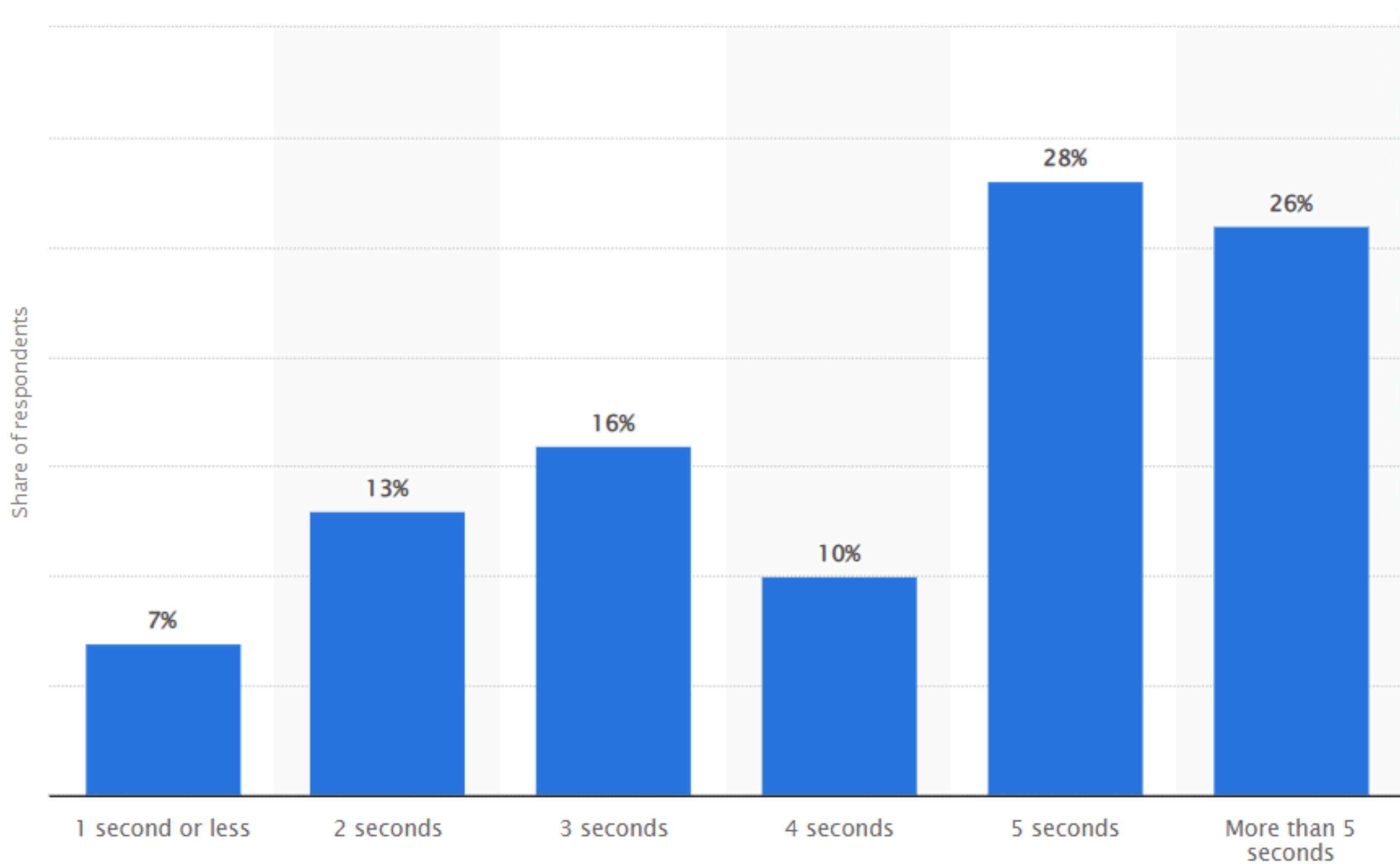
"It's Too Expensive and Time-Consuming"

"We Don't Have Enough Data to Justify It"

"We'll Handle It Later"

Why Performance Matters?

User Experience and Retention



Conversion Rates

	Optimized Page	Default Page	Δ
DCL	4.05s	3.52s	+15%
LCP	5.7s	8.3s	-31%
CR Lead/Visits			+15%
CR Cart views/Visits			+11%
Sales			+8%

SEO Rankings

Google uses page speed as a ranking factor. Faster websites rank higher, get more visibility, and drive more organic traffic.

Greater reach

Performance matters for users on slower networks or older devices. A lightweight, optimized site ensures everyone can access your content.

What is considered good performance?

Google Core Web Vitals

They are a set of performance metrics that measure the user experience of a webpage.

Largest Contentful Paint (LCP)

Measures how quickly the largest visible content element (e.g., an image, video, or block of text) loads. An ideal LCP is less than 2.5 seconds.

Interaction To Next Paint (INP)

Measures responsiveness. To provide a good user experience, strive to have an INP of less than 200 milliseconds.

Cumulative Layout Shift (CLS)

Measures visual stability. To provide a good user experience, strive to have a CLS score of less than 0.1.

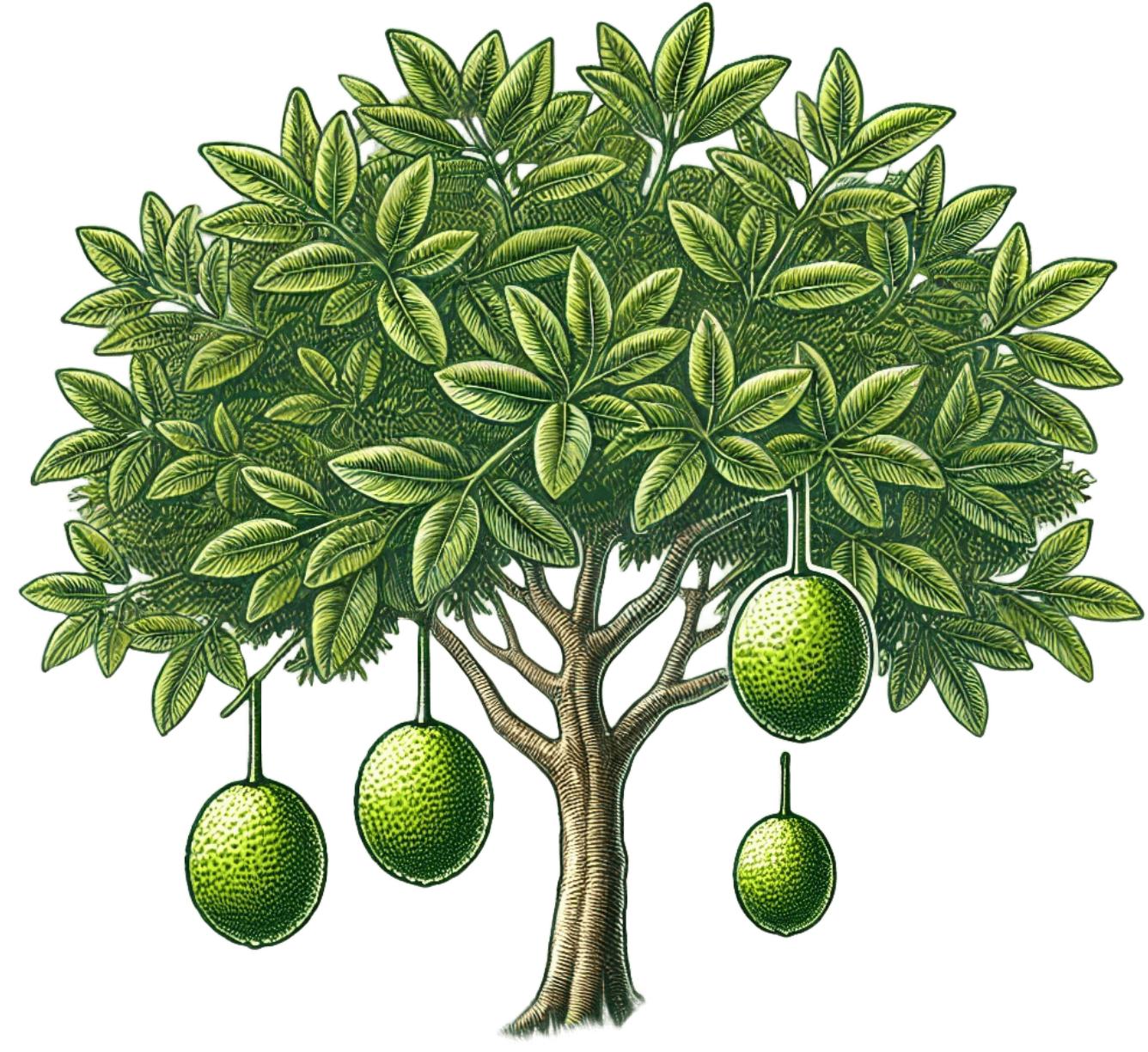
In Summary

Largest Contentful Paint (LCP):	< 2.5s
Cumulative layout shift	< 0.1 (i.e < 10% shift)
Interaction to next paint (ink)	< 200ms
Time to first byte (TTFB)	< 200 ms
First Contentful Paint (FCP)	< 1.8 s

What is our goal today?

Target low hanging fruit

- 1.Optimizing Critical Rendering Path
2. Reducing Network Payload



Critical Rendering Path

At a high level, your browser's rendering engine performs the following steps:

- Construct DOM + CSSOM (resource fetching)
- Create Render tree (DOM + CSSOM)
- Generate layout
- Paint the content to the browser

Resource Fetching

Resource hints and script attributes:

- preload: Preloads critical resources needed for the page to render, improving performance by fetching them early.
- prefetch: Downloads resources that are likely to be needed soon, typically for subsequent navigation or user actions.
- defer: Downloads the script in parallel to HTML parsing but defers execution until after the HTML document has been completely parsed.
- async: Downloads the script in parallel to HTML parsing. Executes the script as soon as it's downloaded, without waiting for HTML parsing to complete.

```
<!-- index.html - No resource hints -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Optimizing Web Performance</title>
  <link href="fonts/Roboto.woff2" type="font/woff2" crossorigin="anonymous">
  <link href="large-image.jpg">
  <link rel="stylesheet" href="styles.css">
  <script src="app.js"></script>
  <script src="https://example-analytics.com/analytics.js"></script>
</head>
<body>
  <h1>Web Performance Optimization</h1>
</body>
</html>
```

```
<!-- index.html - With resource hints -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Optimizing Web Performance</title>

  <!-- Preload critical resources (font, main script, etc.) -->
  <link rel="preload" as="font" href="fonts/Roboto.woff2" type="font/woff2" crossorigin>

  <!-- Prefetch resources likely needed in the future -->
  <link rel="prefetch" as="image" href="large-image.jpg">

  <link rel="stylesheet" href="styles.css">

  <!-- defer execution until html is parsed -->
  <script defer src="app.js"></script>

  <!-- download and run as soon as ready -->
  <script async src="https://example-analytics.com/analytics.js"></script>
</head>
<body>
  <h1>Web Performance Optimization</h1>
</body>
</html>
```

```
<!-- index.html - With resource hints -->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Optimizing Web Performance</title>

  <!-- Preload critical resources (font, main script, etc.) -->
  <link rel="preload" as="font" href="fonts/Roboto.woff2" type="font/woff2" crossorigin>

  <!-- Prefetch resources likely needed in the future -->
  <link rel="prefetch" as="image" href="large-image.jpg">

  <link rel="stylesheet" href="styles.css">

  <!-- defer execution until html is parsed -->
  <script defer src="app.js"></script>

  <!-- download and run as soon as ready -->
  <script async src="https://example-analytics.com/analytics.js"></script>
</head>
<body>
  <h1>Web Performance Optimization</h1>
</body>
</html>
```

Preventing Layout shift

Use Loading skeletons or other placeholders

```
// Lazy load the UserProfile component
const UserProfile = React.lazy(() => import("./UserProfile"));

const App = () => {
  return (
    <div className="app-container">
      <h1>User Profile</h1>
      <Suspense fallback={<SkeletonLoader />}>
        <UserProfile />
      </Suspense>
    </div>
  );
};

// Skeleton Loader for Suspense fallback
const SkeletonLoader = () => (
  <div className="profile-skeleton">
    <div className="skeleton skeleton-avatar"></div>
    <div className="skeleton skeleton-name"></div>
    <div className="skeleton skeleton-bio"></div>
  </div>
);
```

Ant Design

Type keywords...

Design Development Components Blog Resources 5.22.5

Divider

A divider line separates different content.

Import `import { Divider } from "antd";`
Source [components/divider](#)
Docs [Edit this page](#) [Changelog](#)

When To Use

- Horizontal
- Divider with title
- Text without h...
- Vertical
- Variant

API

Design Token

Components Overview

General

Button

FloatButton 5.0.0

Icon

Typography

Layout

Divider

Flex 5.10.0

Grid

Layout

Space

Splitter 5.21.0

Navigation

Anchor

Breadcrumb

Ant Design

Type keywords...

Design Development Components Blog Resources 5.22.5

Divider

A divider line separates different content.

Import `import { Divider } from "antd";`
Source [components/divider](#)
Docs [Edit this page](#) [Changelog](#)

When To Use

- Divide sections of an article.
- Divide inline text and links such as the operation column of table.

Examples

Flex 5.10.0

Grid

Layout

Space

Splitter 5.21.0

Navigation

Anchor

Text

Left Text

https://ant.design/components/overview

Ant Design Type keywords... Design Development Components Blog Resources 5.22.5 En A

Components Overview

antd provides plenty of UI components to enrich your web applications, and we will improve components experience consistently. We also recommend some great [Third-Party Libraries](#) additionally.

Search in components

General [4]

- Button
- FloatButton 5.0.0
- Icon
- Typography

Layout [6]

- Divider
- Flex 5.10.0
- Grid
- Layout
- Space
- Splitter 5.21.0

Navigation

Anchor

Breadcrumb

Dropdown

Menu

Pagination

Components Overview

General

Button

FloatButton 5.0.0

Icon

Typography

Layout

Divider

Flex 5.10.0

Grid

Layout

Space

Splitter 5.21.0

Navigation

Anchor

Breadcrumb

Dropdown

Menu

Pagination

Components Overview

General

Button

FloatButton 5.0.0

Icon

Typography

Layout

Divider

Flex 5.10.0

Grid

Layout

Space

Splitter 5.21.0

Navigation

Anchor

Breadcrumb

Dropdown

Menu

Pagination

Components Overview

General

Button

FloatButton 5.0.0

Icon

Typography

Layout

Divider

Flex 5.10.0

Grid

Layout

Space

Splitter 5.21.0

Navigation

Anchor

Breadcrumb

Dropdown

Menu

Pagination

Reducing Network Payload

- Lazy load components to reduce initial bundle size.
- Lazy load assets.
- Optimize assets.

Lazy loading components

```
// App.tsx - No code-splitting
import React from 'react';
import LargeComponent from './LargeComponent';
import OtherComponent from './OtherComponent';

const App: React.FC = () => (
  <div>
    <LargeComponent />
    <OtherComponent />
  </div>
);

export default App;
```

```
// App.tsx - With code-splitting using React.lazy
import React, { Suspense, lazy } from 'react';
const LargeComponent = lazy(() => import('./LargeComponent'));
const OtherComponent = lazy(() => import('./OtherComponent'));

const App: React.FC = () => (
  <Suspense fallback={<div>Loading...</div>}>
    <LargeComponent />
    <OtherComponent />
  </Suspense>
);

export default App;
```

Lazy loading images

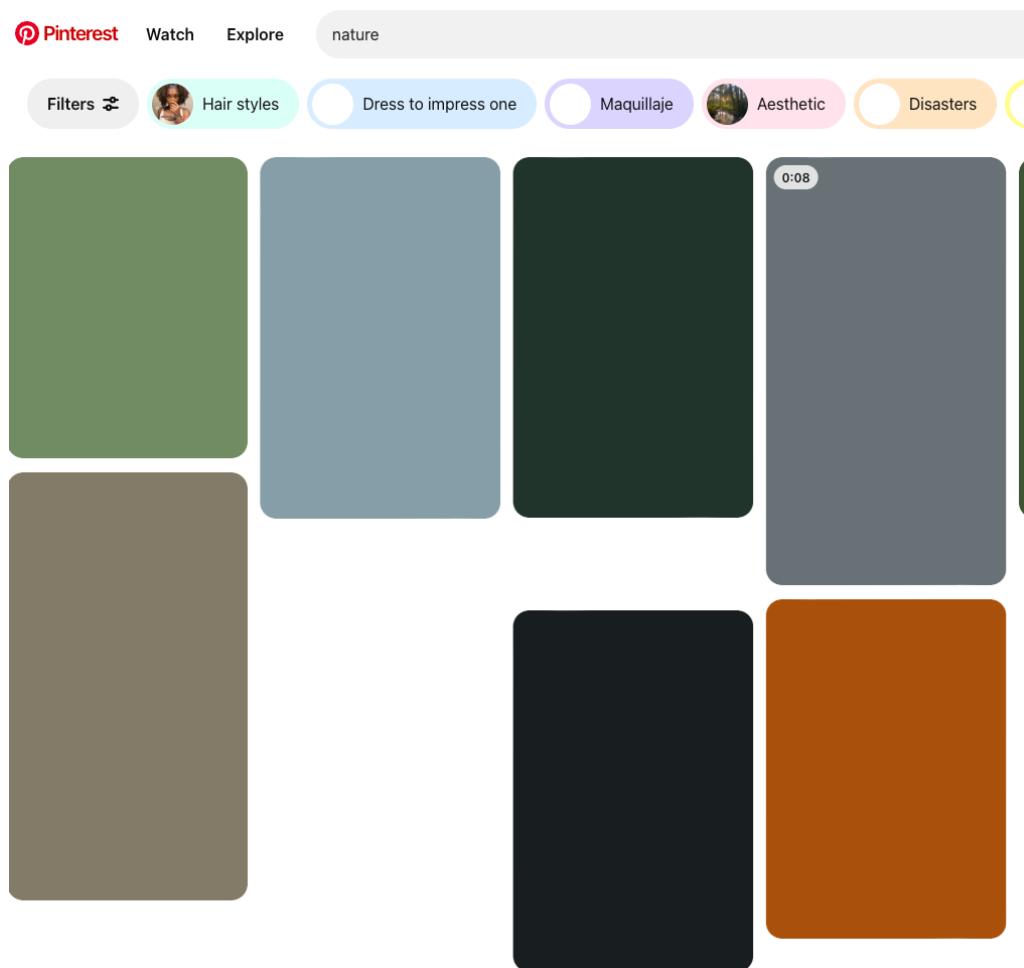
Lazy-loading images with JS

Common Pattern

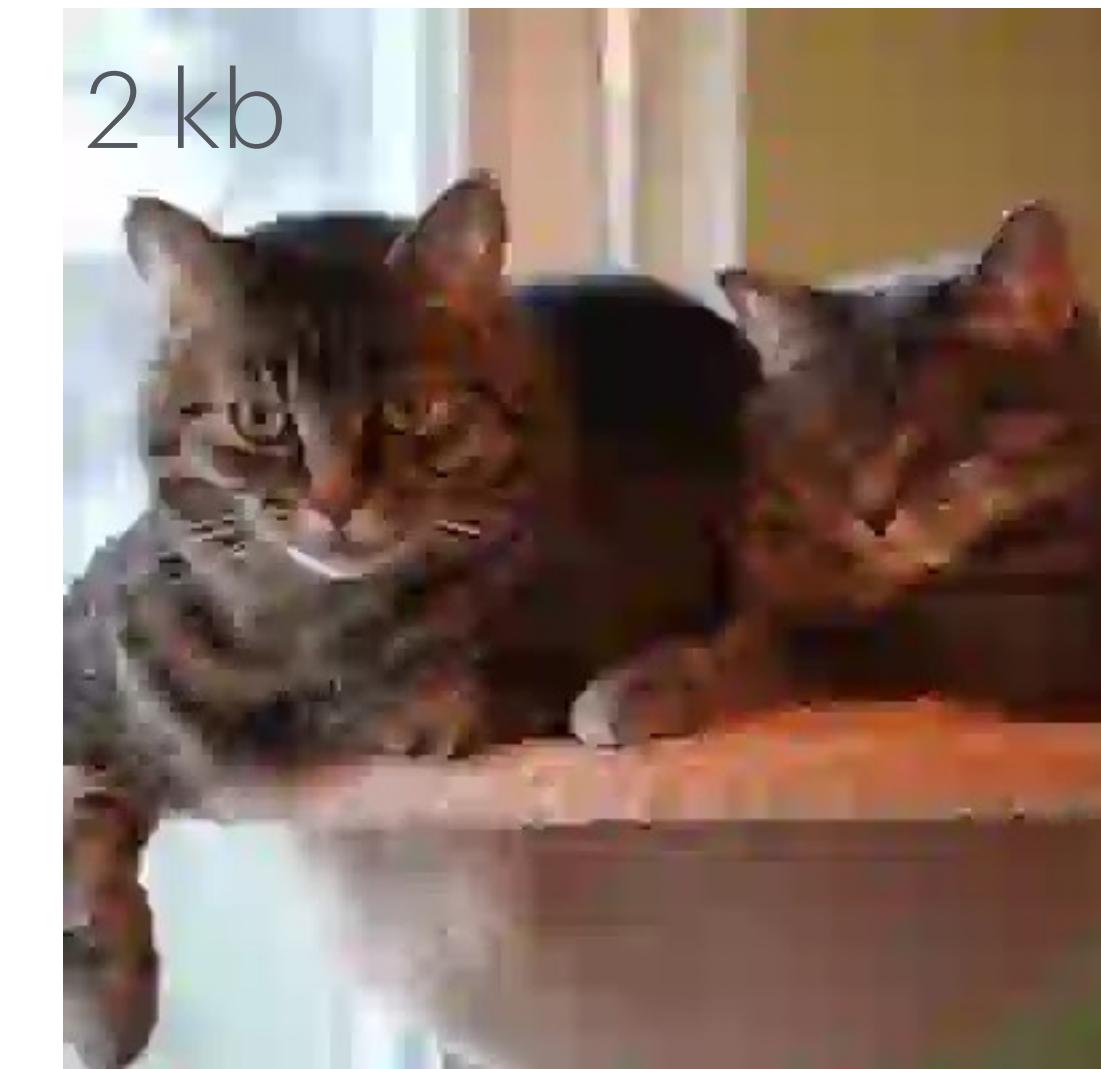
```
  

```

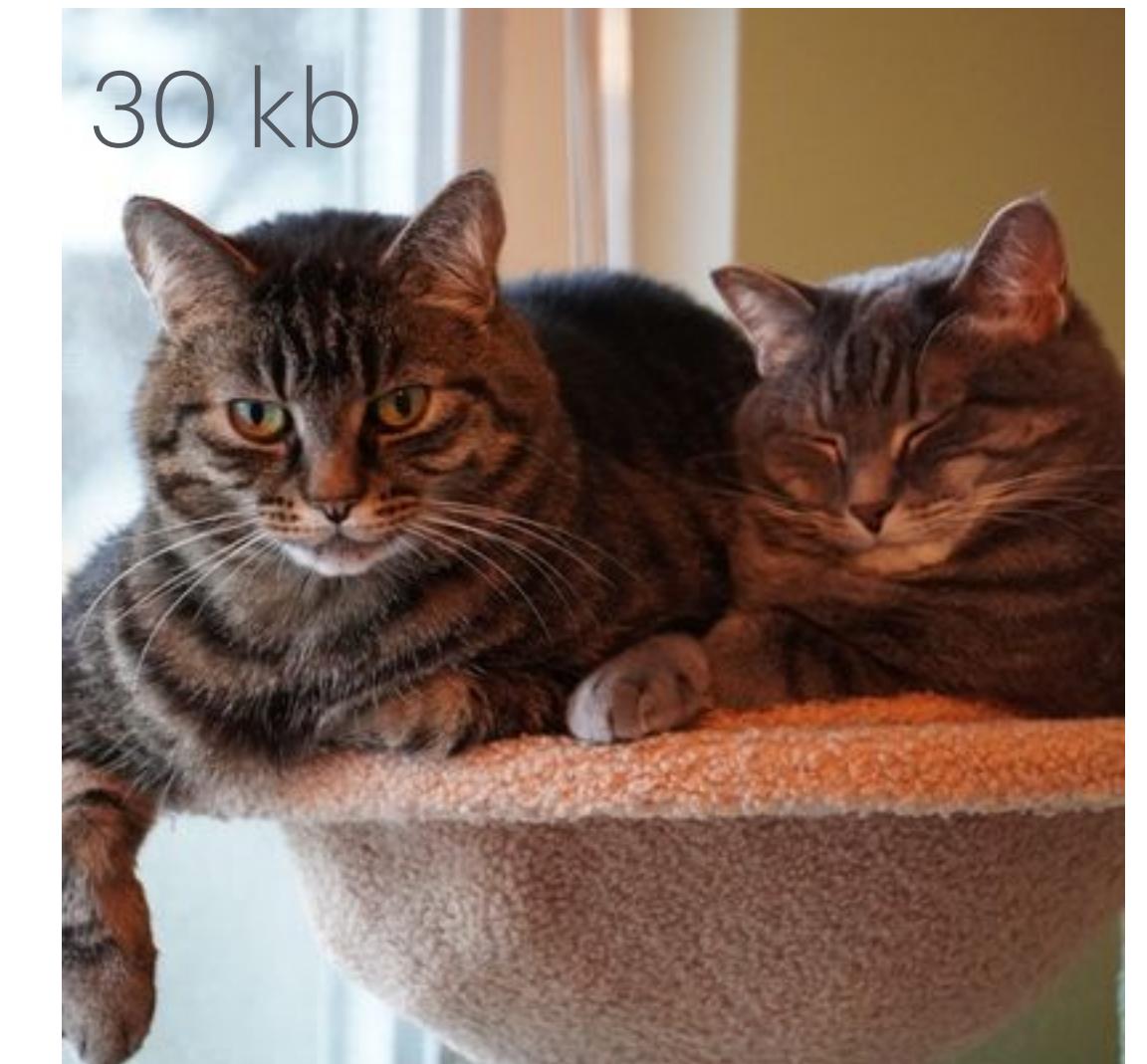
Progressive loading



Show dominant colours



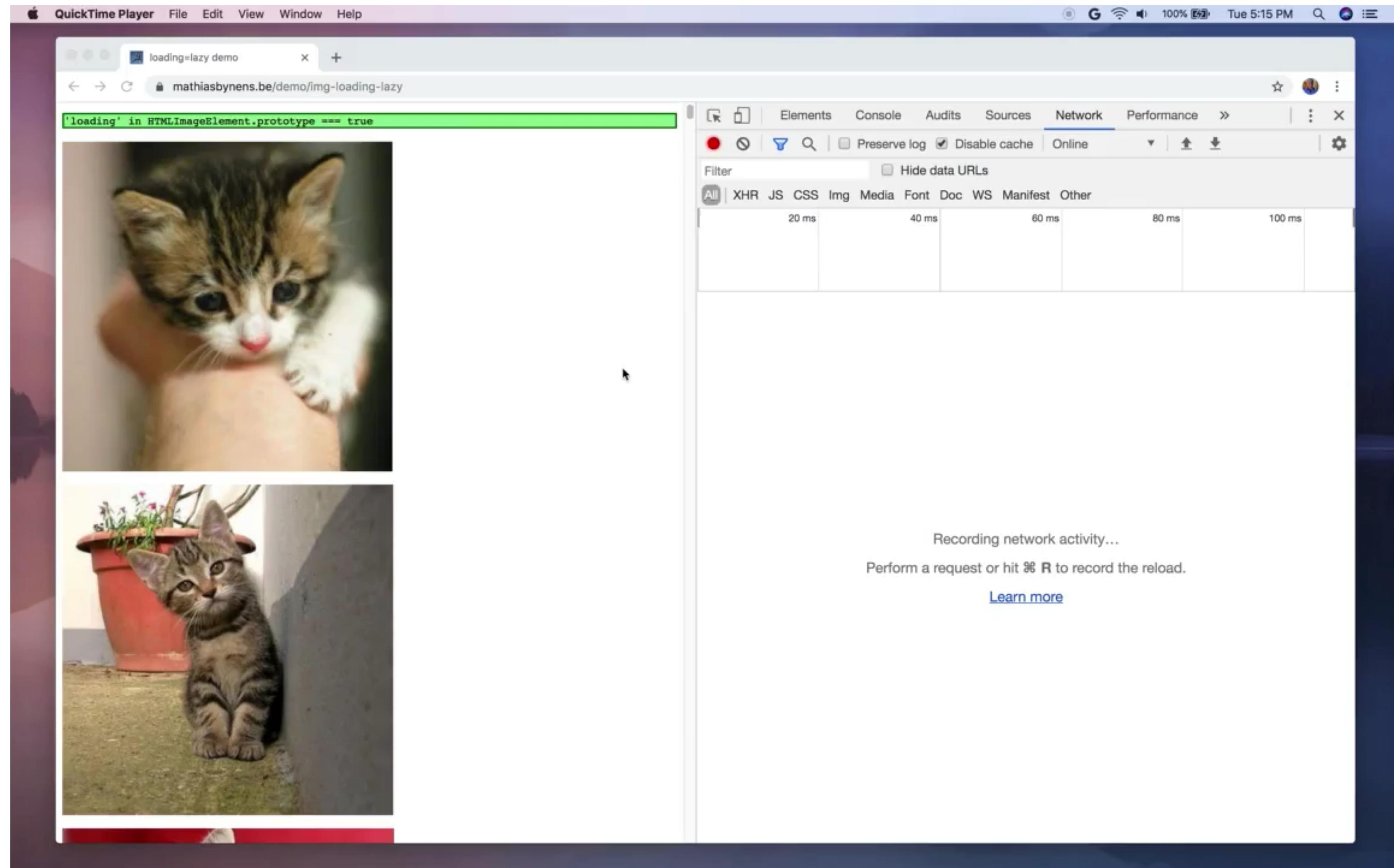
Low Quality Image Placeholders (LQIP)



Native browser support

```

```



Optimizing images

Use CDN image optimisation

...cat.jpg?q=50&h=500

Set quality level to 50

Resize to height of 500, determine width automatically

Popular Image CDNs

Akamai | Cloudinary | Cloudflare Images | Imgix

/i1.jpg
464kB | 2000 x 1334

Name	X	Headers	Preview	Response	>>
i1.jpg					
1 / 2 requests	464 kB	2000 x 1334	1000:667	image/jpeg	

/i1.jpg?h=500
61.3kB | 749 x 500

Name	X	Headers	Payload	Preview	>>
i1.jpg?h=500					
1 / 2 requests	61.3 kB	749 x 500	749:500	image/jpeg	

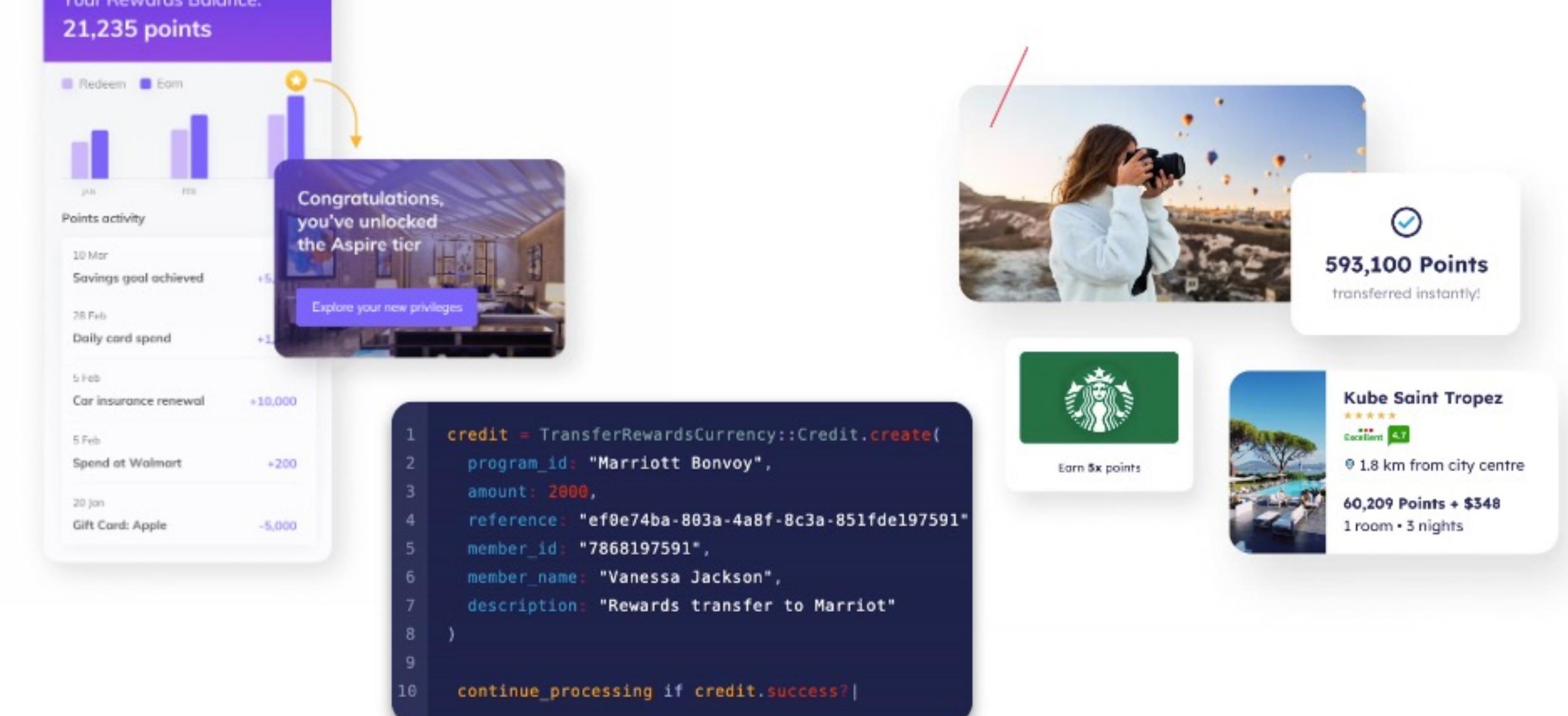
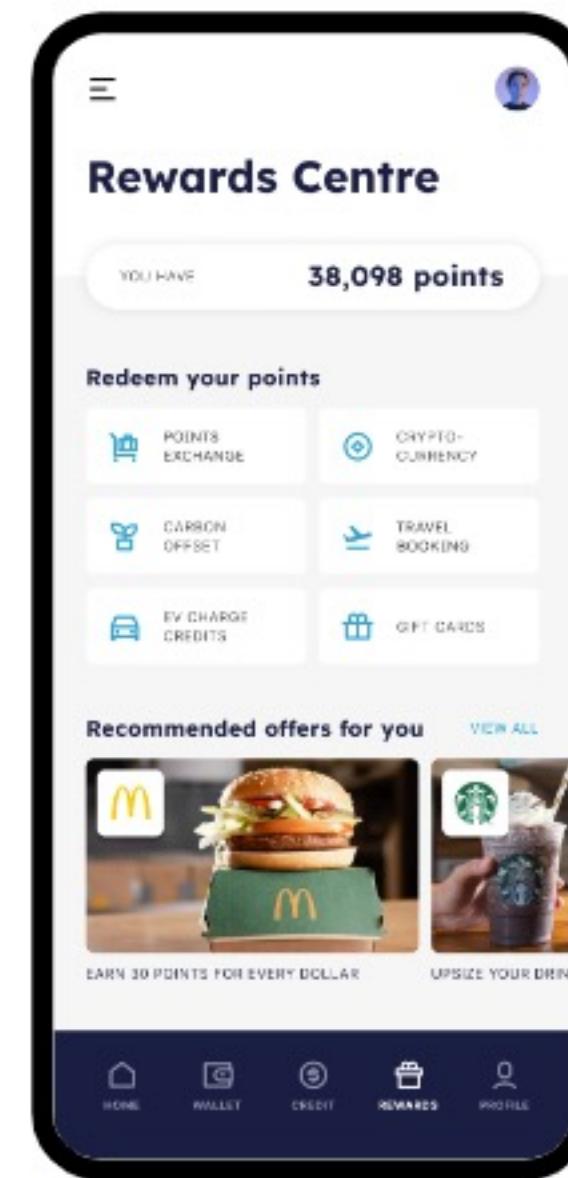
85% reduced size

Key Takeaways

1. Use data to start the conversation with your stakeholders
2. Use resource hints to optimise critical rendering path (<link rel="preload" as="font"... />)
3. Use lazy loading for non-immediate components (lazy(() => import('./LargeComponent')))
4. Use lazy loading for images and other assets ()
5. Use a CDN to optimise images for your app
6. Use effective loaders like skeletons to mitigate layout shift

Ascenda

Accelerate your growth ...
... with world-class rewards



```
1 credit = TransferRewardsCurrency::Credit.create(
2   program_id: "Marriott Bonvoy",
3   amount: 2000,
4   reference: "ef0e74ba-803a-4a8f-8c3a-851fde197591"
5   member_id: "7868197591",
6   member_name: "Vanessa Jackson",
7   description: "Rewards transfer to Marriot"
8 )
9
10 continue_processing if credit.success?|
```

Hi there!

Let's make loyalty
rewarding, together

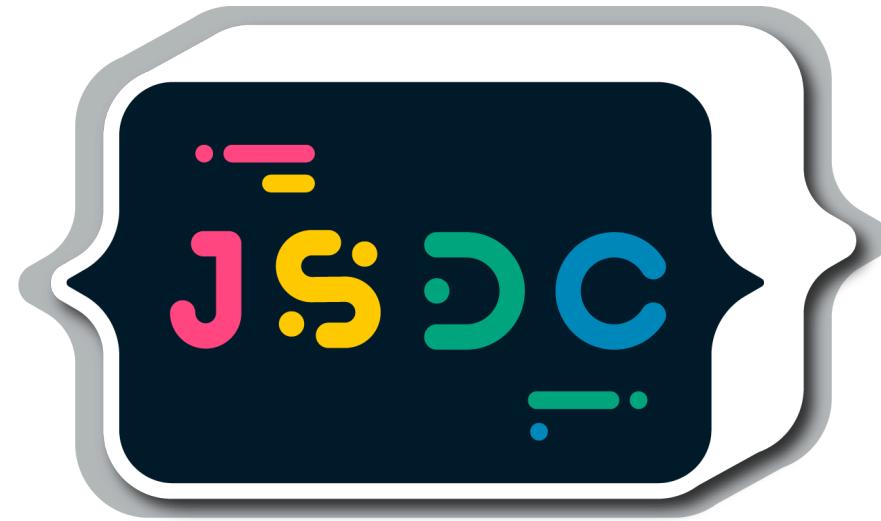


careers.ascendaloyalty.com

linkedin.com/company/ascenda-loyalty



Thank you



SLIDES👉 <https://bit.ly/jsdctw-2024-web-perf>

Dhrubajit Paul
Chowdhury
@dhrubajit
Ascenda

