DEVOCEAN

웹 프론트엔드 성능 최적화 적용하기

프론트엔드 개발자 장세영, 이유진, 임수현

목차

01 성능 최적화의 필요성

01 성능 최적화, 왜 필요할까?

'성능이 저하되면 사용자가 떠나고 매출이 감소한다'
=> 성능이 향상되면 그만큼 사용자가 늘고 매출이 오른다!

📶 페이지 표시 시간 증가에 따른 이탈률

페이지 표시 시간이 1초에서 3초로 느려진 경우, 사용자 이탈률 32% 증가

페이지 표시 시간이 1초에서 5초로 느려진 경우, 사용자 이탈률 90% 증가

페이지 표시 시간이 1초에서 6초로 느려진 경우, 사용자 이탈률 106% 증가

페이지 표시 시간이 1초에서 10초로 느려진 경우, 사용자 이탈률 123% 증가

핀터레스트
로딩시간 40% 감소 -> 검색 유입률 및 가입자수 15%증가

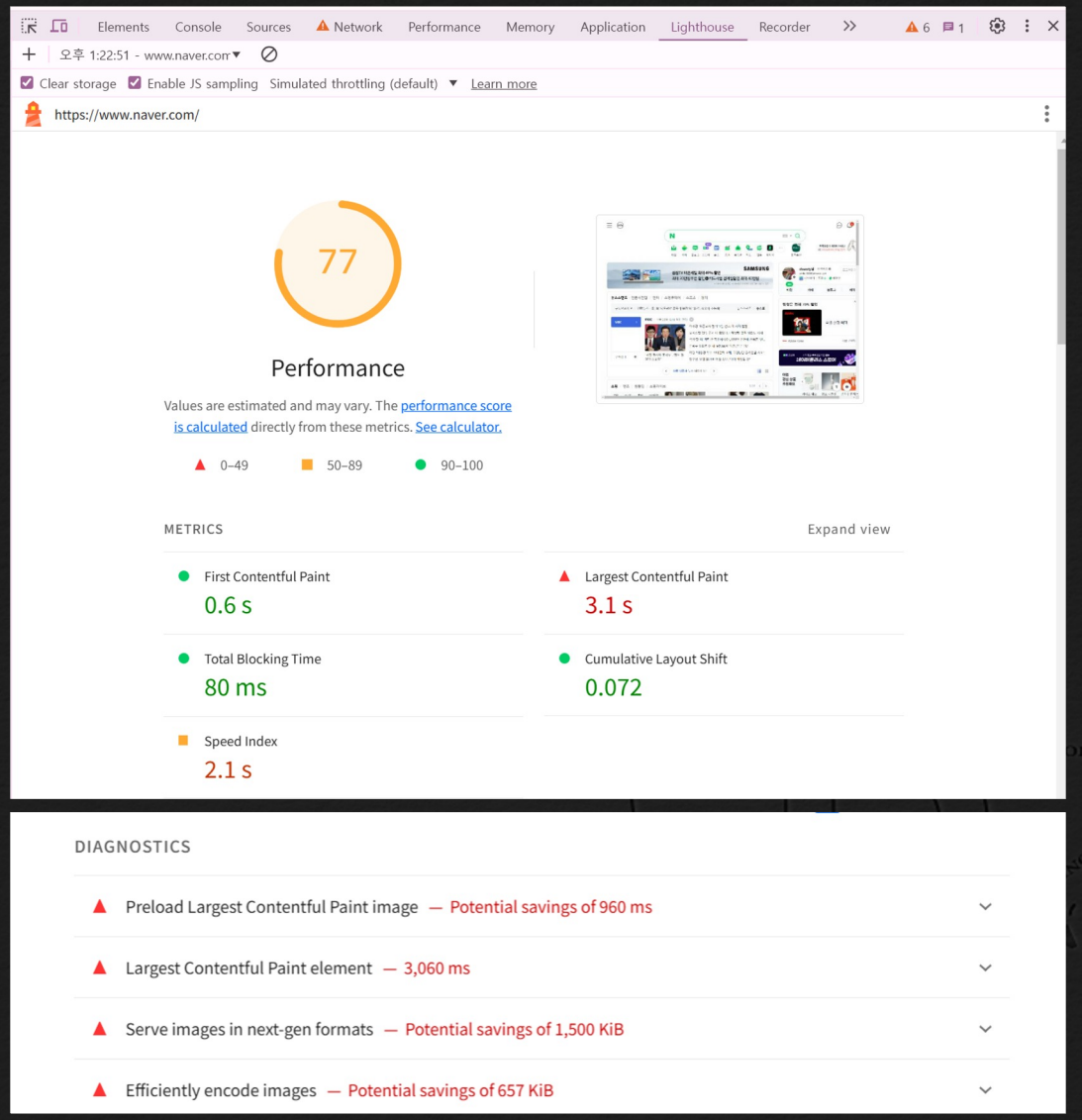
COOK
평균 페이지 로드 시간 850ms 감소->
세션당 페이지 조회수 10%증가 , 이탈률 7%감소

02 성능 측정 도구 및 항목

02 성능 측정 도구 및 항목 - 성능 측정 도구

Lighthouse

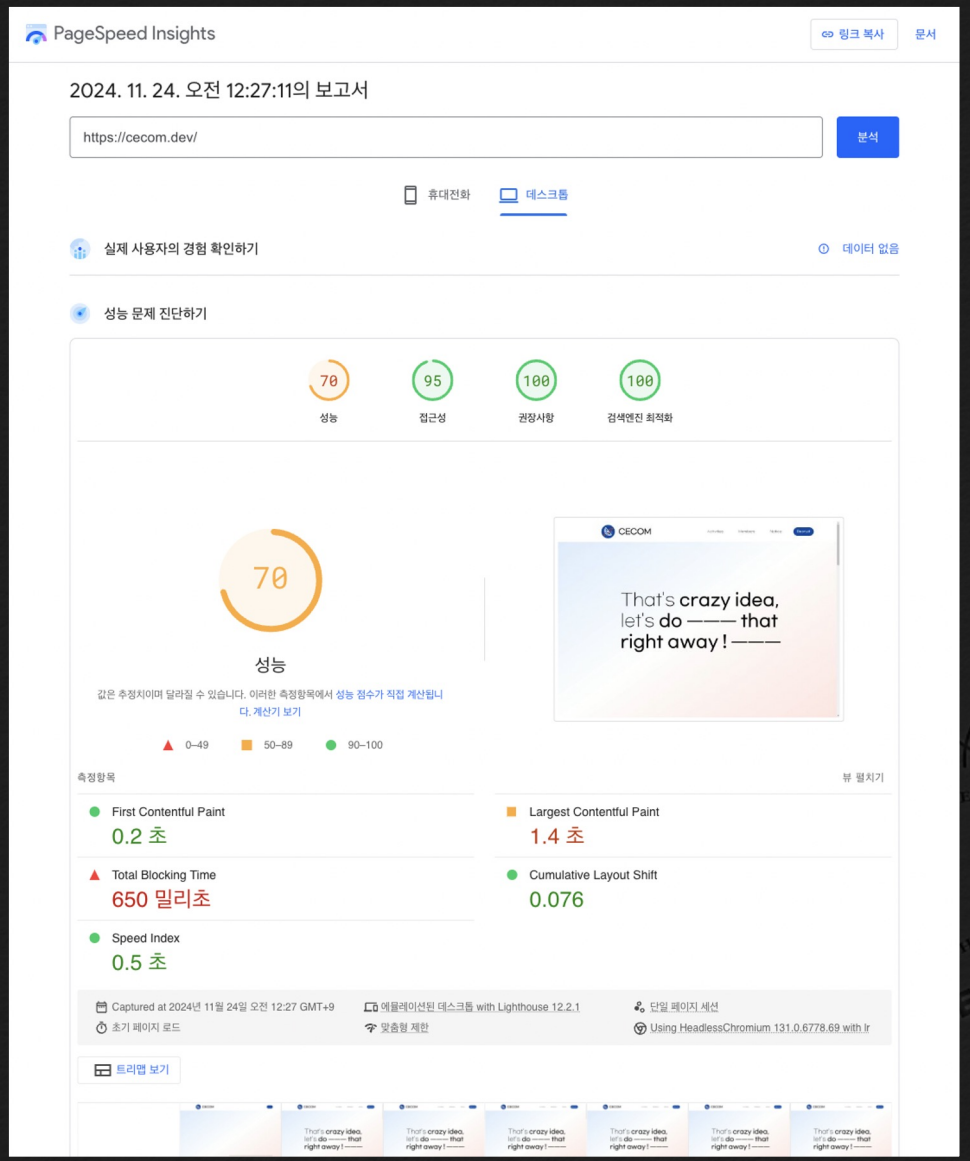
- Google Chrome에서 제공하는 자동화 도구
- Performance, Accessibility, Best Practices, SEO, PW 평가 지표를 이용해 점수 계산
- 웹사이트를 분석하고 개선 사항 제안
- 로컬 개발 환경에서 테스트 가능



02 성능 측정 도구 및 항목 - 성능 측정 도구

PageSpeed Insights

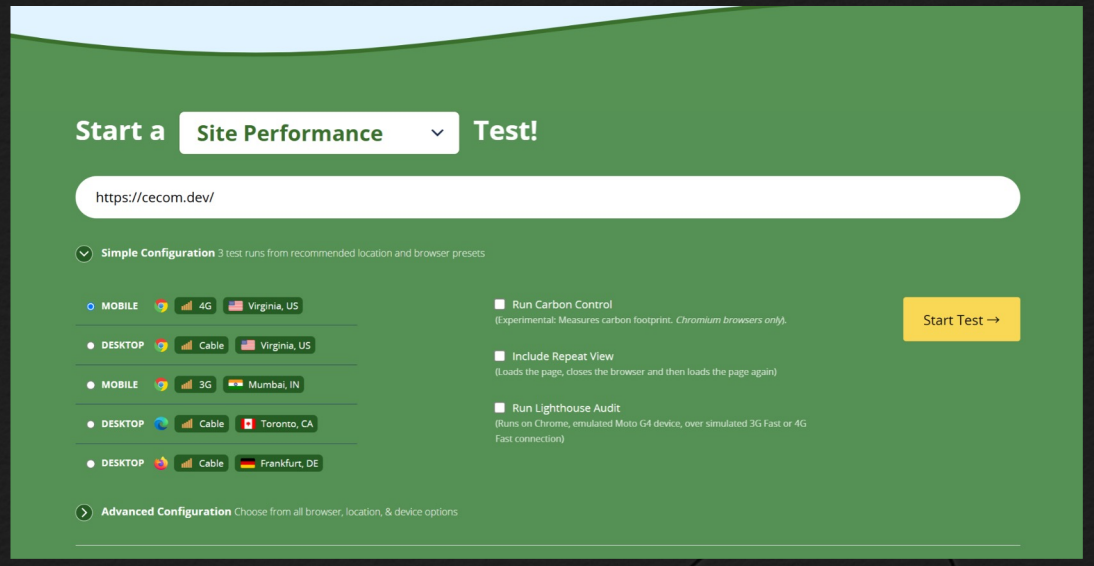
- Lighthouse를 기반으로 Google이 제공하는 웹 성능 분석 도구
- 모바일 및 데스크톱 점수 분리 제공
- 실제 사이트 데이터를 확인 가능
- Core Web Vitals 지표 제공

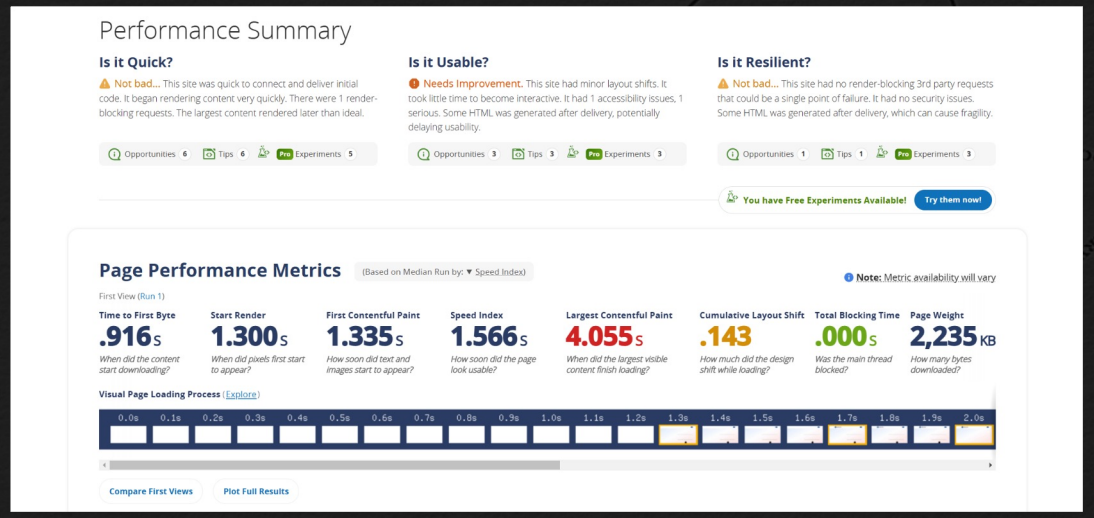


02 성능 측정 도구 및 항목 - 성능 측정 도구

WebPageTest

- 웹 성능 측정 관점에서 여러가지의 지표 정보 제공
- 타겟 환경을 지정해서 측정 가능
- 결과 측정 리포트 다양한 분석





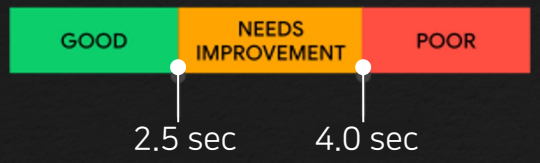
02 성능 측정 도구 및 항목 - 성능 측정 항목

Core Web Vitals
구글이 웹사이트 성능 최적화를 위해 중요하다고 판단하는 세 가지 지표

(Loading)

LCP

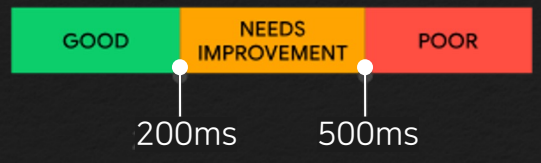
Largest Contentful Paint



(Interactivity)

INP

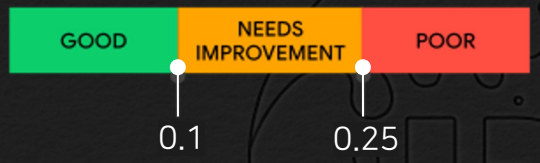
Interaction to Next Paint



(Visual Stability)

CLS

Cumulative Layout Shift



<출처 :web.dev >

02 성능 측정 도구 및 항목 - 성능 측정 항목 (Core Web Vitals)

LCP : 최대 콘텐츠 렌더링 시간

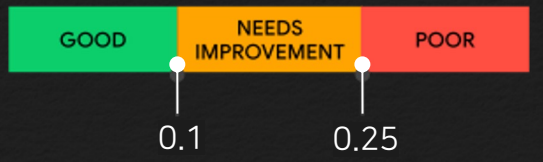
- 로드 성능을 측정
- 페이지가 처음 로드를 시작한 시점을 기준으로 뷰 포트 내에 있는 가장 큰 이미지 또는 텍스트 블록의 렌더링 시간을 측정

02 성능 측정 도구 및 항목 - 성능 측정 항목 (Core Web Vitals)

(Visual Stability)

CLS

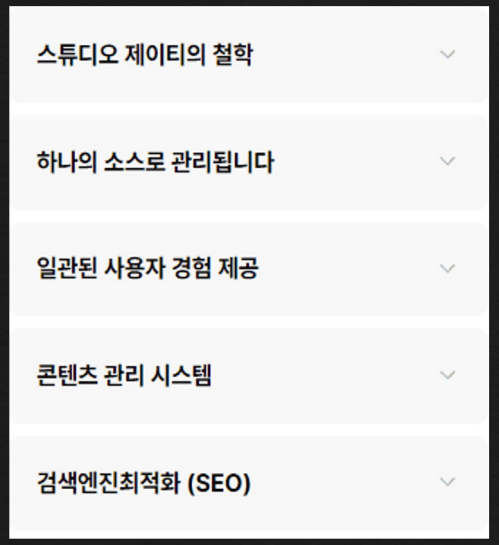
Cumulative Layout Shift



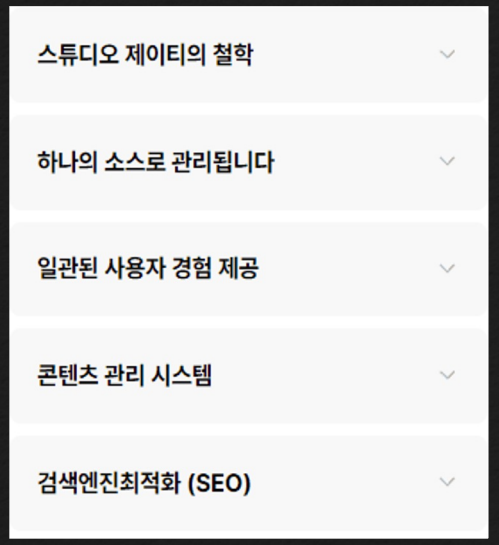
CLS : 누적 레이아웃 변경

- 시각적 안정성을 측정
- 사용자가 예상하지 못한 레이아웃 변경이 얼마나 자주 발생했는지를 측정하는 지표
- 텍스트를 읽거나 버튼을 클릭하려는 순간 콘텐츠가 이동하면 사용자 경험에 부정적인 영향을 줌

02 성능 측정 도구 및 항목 - 성능 측정 항목 (Core Web Vitals)



BAD INP
느린 반응성

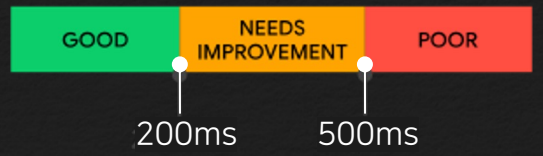


GOOD INP

(Interactivity)

INP

Interaction to Next Paint



INP : 다음 페인트에 대한 상호작용

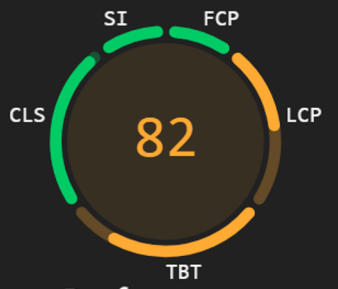
- 상호작용 측정
- 사용자 인터렉션 (클릭, 스크롤, 키 입력 등) 이후 다음 화면이 렌더링될 때까지 걸리는 시간을 측정
- 빠른 INP는 높은 전환율을 의미하고, 이는 더 나은 사용자 경험 제공 가능

출처 : https://studio-jt.co.kr/inp-interaction-to-next-paint/

02 성능 측정 도구 및 항목 - 성능 측정 항목 (Core Web Vitals 외의 항목)

Lighthouse Performance 항목

- Largest Contentful Paint (25%)
- Cumulative Layout Shift (25%)
- First Contentful Paint (10%)
- Speed Index (10%)
- Total Blocking Time (30%)



Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS Expand view

● First Contentful Paint 0.4 s	■ Largest Contentful Paint 2.2 s
■ Total Blocking Time 200 ms	● Cumulative Layout Shift 0.07
● Speed Index 0.8 s	

02 성능 측정 도구 및 항목 - 성능 측정 항목

Lighthouse Performance 항목 점수

- Largest Contentful Paint (25%)
- Cumulative Layout Shift (25%)
- First Contentful Paint (10%)
- Speed Index (10%)
- Total Blocking Time (30%)

First Contentful Paint (10%)

FCP는 사용자가 페이지로 이동한 후 브라우저에서
첫 번째 DOM 콘텐츠를 렌더링하는 데 걸리는 시간을 측정

FCP 시간 (초)	색 구분
0~1.8	녹색 (빠름)
1.8~3	주황색 (보통)
3명 초과	빨간색 (느림)

02 성능 측정 도구 및 항목 - 성능 측정 항목

Lighthouse Performance 항목 점수

- Largest Contentful Paint (25%)
- Cumulative Layout Shift (25%)
- First Contentful Paint (10%)
- Speed Index (10%)
- Total Blocking Time (30%)

Speed Index (10%)

페이지 로드 중 콘텐츠가 시각적으로 표시되는 속도를 측정

속도 지수 (초)	색 구분
0~3.4	녹색 (빠름)
3.4~5.8	주황색 (보통)
5.8 이상	빨간색 (느림)

02 성능 측정 도구 및 항목 - 성능 측정 항목

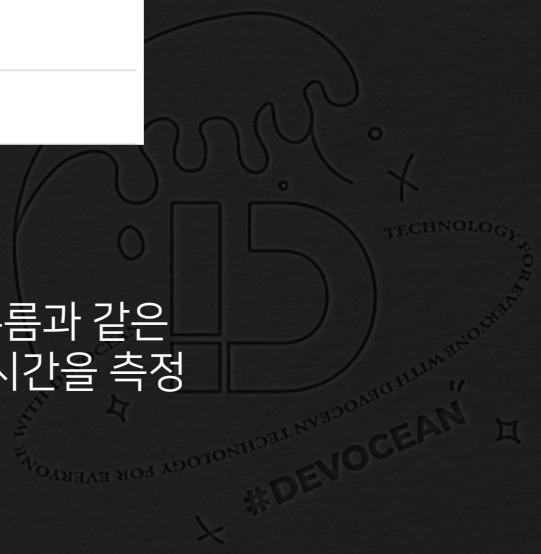
Lighthouse Performance 항목 점수

- Largest Contentful Paint (25%)
- Cumulative Layout Shift (25%)
- First Contentful Paint (10%)
- Speed Index (10%)
- Total Blocking Time (30%)

TBT 시간 (밀리초 단위)	색 구분
0~200	녹색 (빠름)
200-600	주황색 (보통)
600명 이상	빨간색 (느림)

Total Blocking Time (30%)

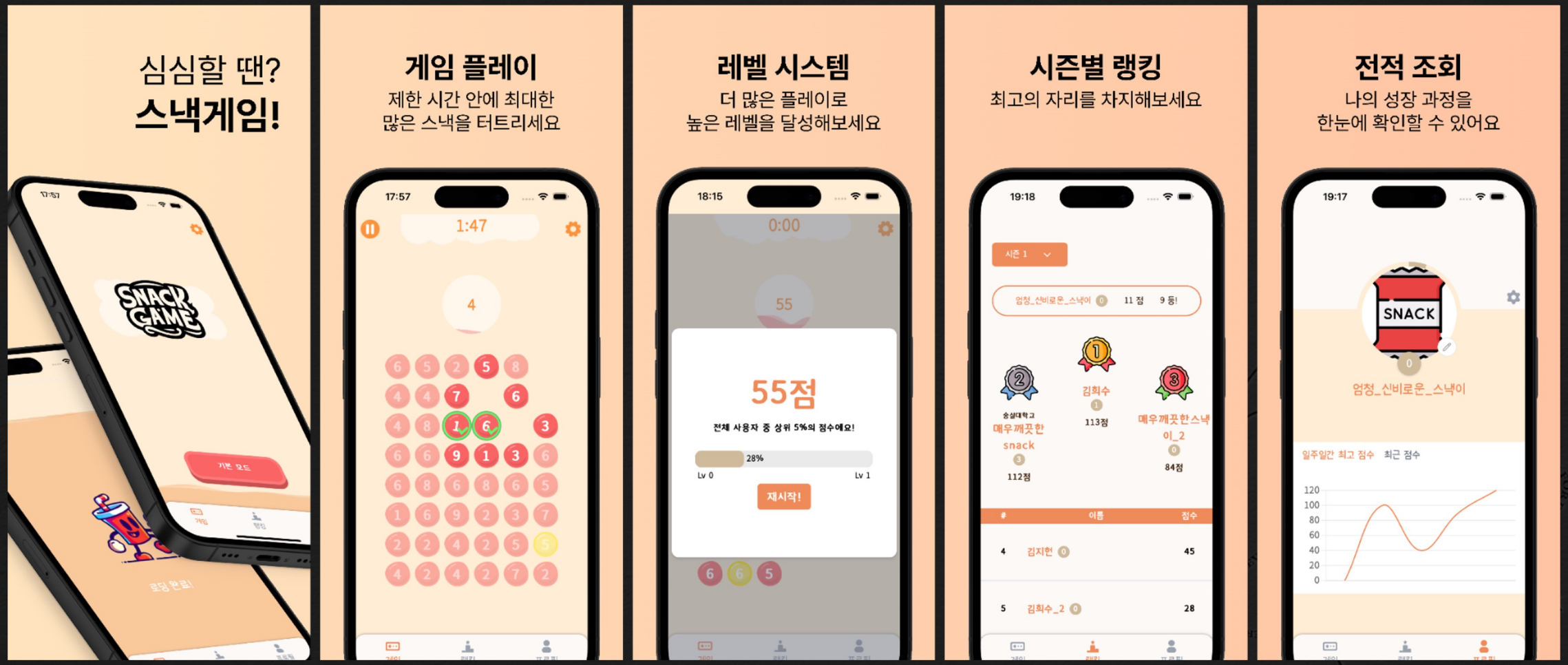
페이지가 마우스 클릭, 화면 탭 또는 키보드 누름과 같은 사용자 입력에 응답하지 못하도록 차단된 총 시간을 측정



03 이미지 최적화

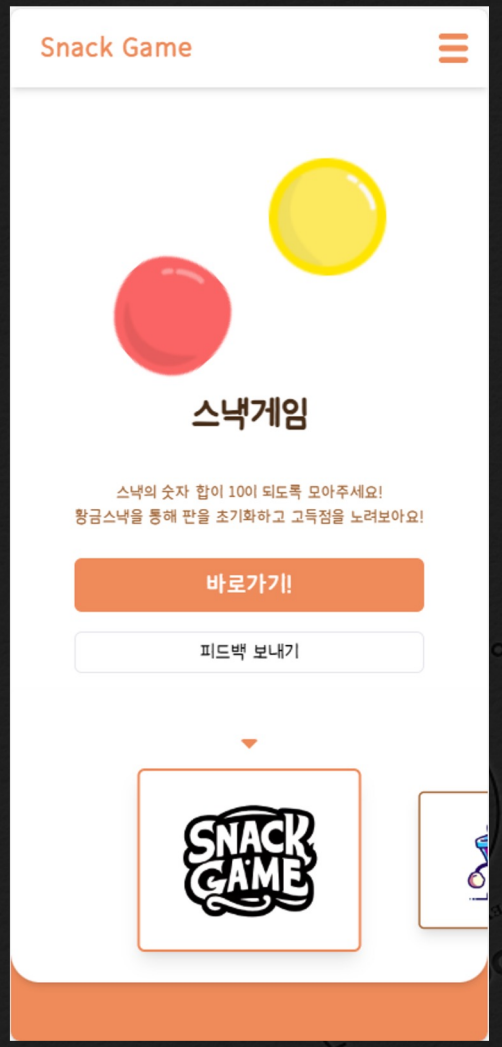
03 이미지 최적화 - 프로젝트 소개: 스낵게임

- 서비스 소개



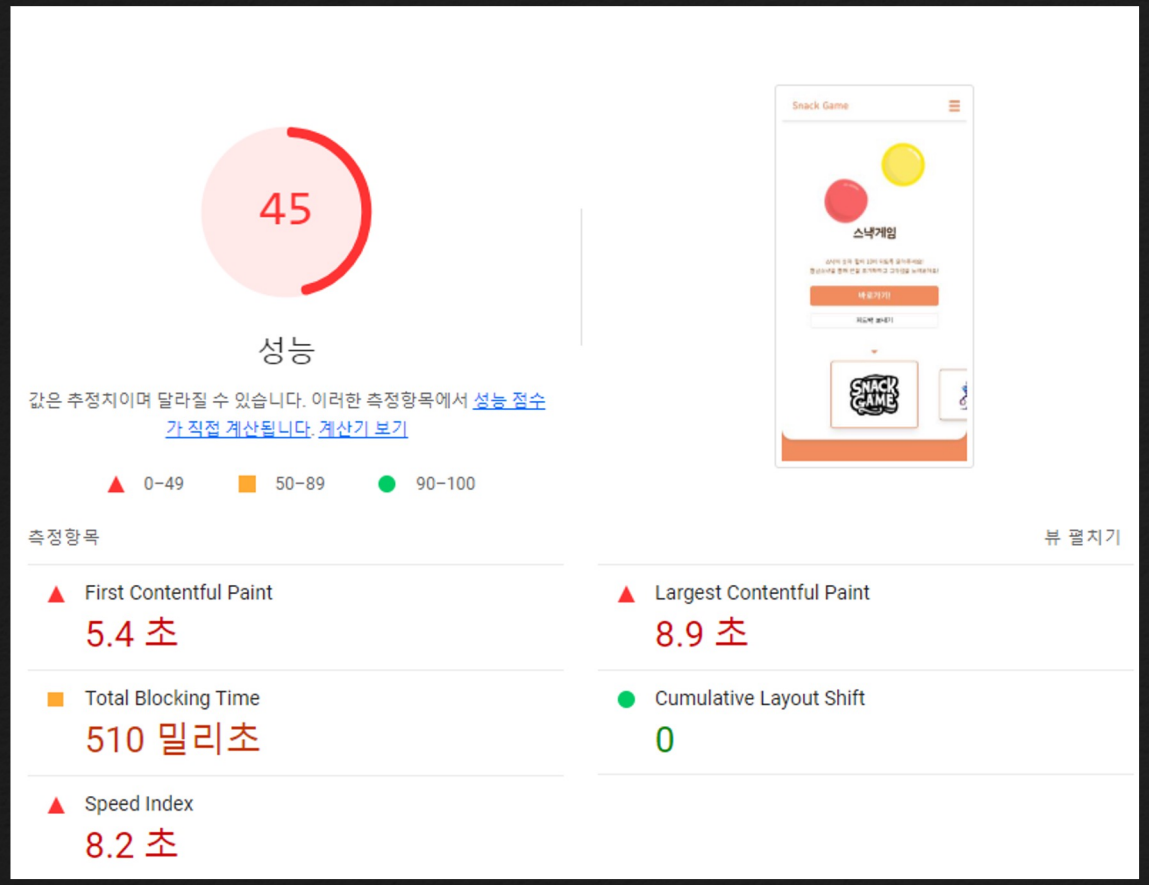
03 이미지 최적화 - 프로젝트 소개: 스낵게임

- 랜딩 페이지
  - 게임 로고, 개발 블로그, 팀원 소개 등 10개 이상의 이미지 사용



03 이미지 최적화 - 프로젝트 소개: 스낵게임

- 랜딩 페이지 개선 전 점수

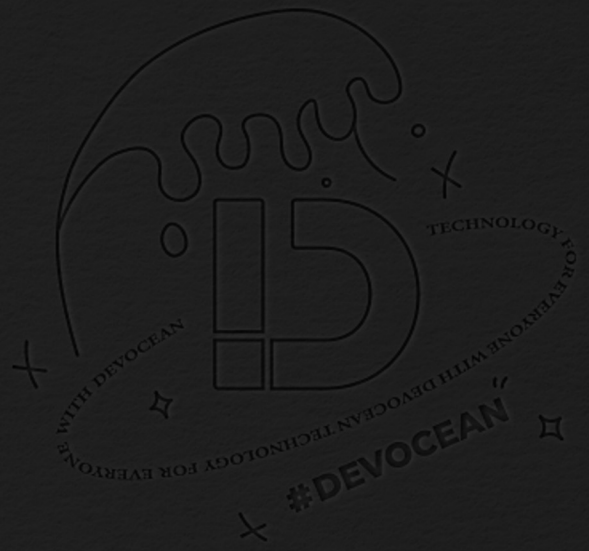


03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?

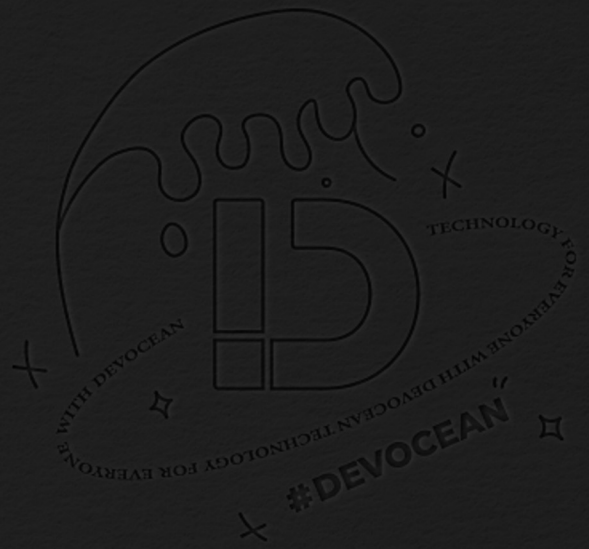
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
    - 보통 이미지는 다른 리소스보다 더 많은 대역폭을 차지



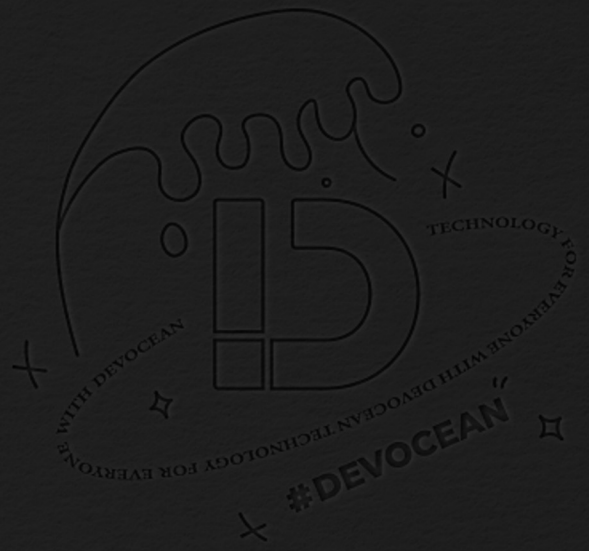
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
    - 보통 이미지는 다른 리소스보다 더 많은 대역폭을 차지
    - 화면에 있는 모든 이미지를 한번에 로드하지 말고, 이미지를 보여줘야 할 때 로드하자



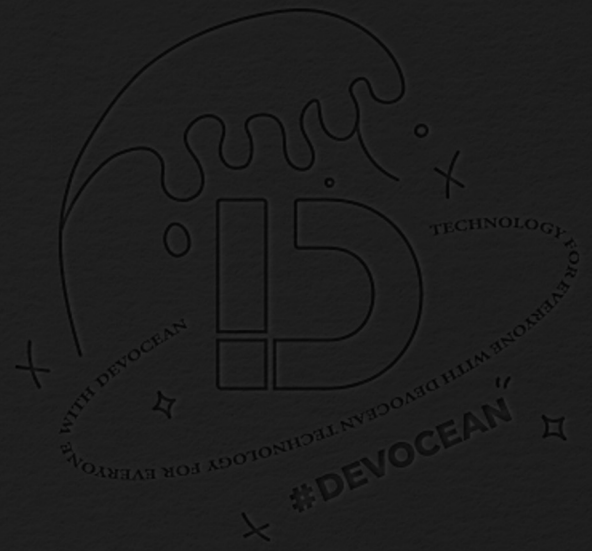
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?



03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
    - 이미지 파일의 크기가 화면에서의 크기보다 과하게 클 경우, 대역폭 낭비와 로딩 속도 저하

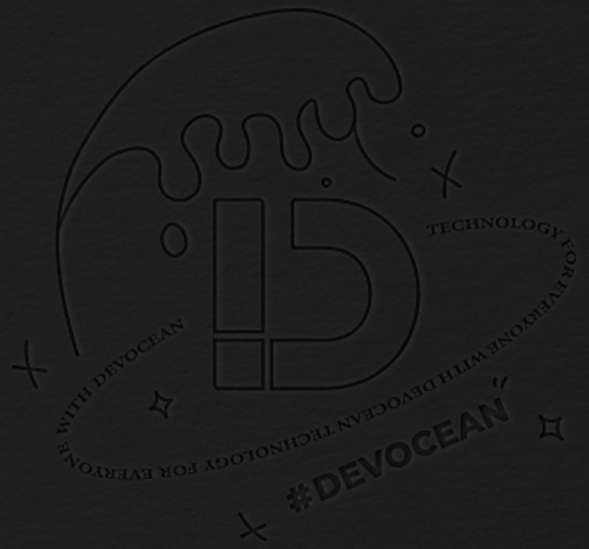


03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
    - 이미지 파일의 크기가 화면에서의 크기보다 과하게 클 경우, 대역폭 낭비와 로딩 속도 저하
    - 실제로 표시되는 크기를 고려해서 이미지 파일의 크기를 조절하자

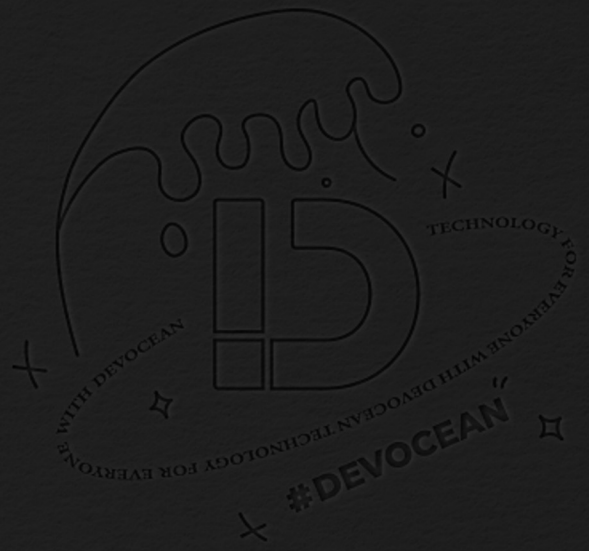
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?



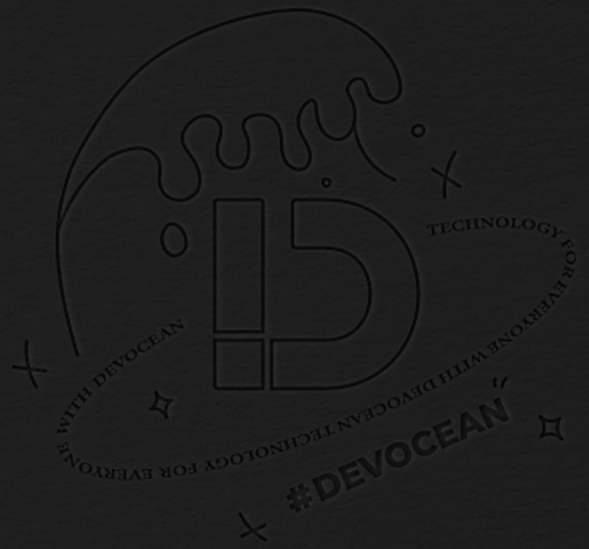
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
    - WebP, AVIF 같은 최신 파일 형식은 기존의 PNG, JPEG보다 더 작은 파일 크기를 제공



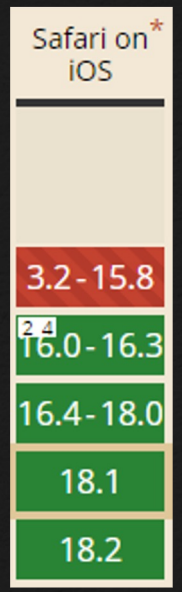
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
    - WebP, AVIF 같은 최신 파일 형식은 기존의 PNG, JPEG보다 더 작은 파일 크기를 제공
    - 브라우저 호환성을 위해 대체 이미지를 제공하자



03 이미지 최적화 - 기법 소개

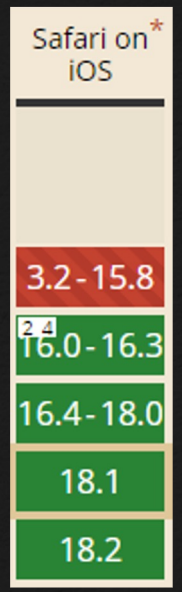
1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
    - WebP, AVIF 같은 최신 파일 형식은 기존의 PNG, JPEG보다 더 작은 파일 크기를 제공
    - 브라우저 호환성을 위해 대체 이미지를 제공하자



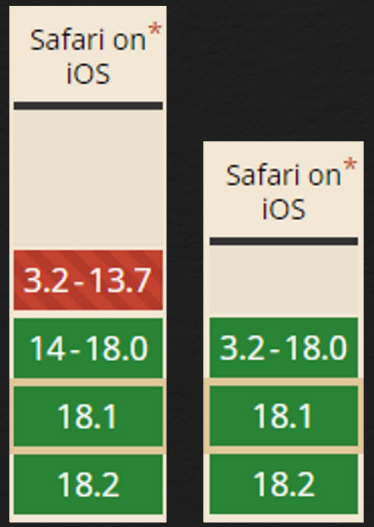
⬅ iOS 16 미만 기기는 AVIF 미지원

03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
    - WebP, AVIF 같은 최신 파일 형식은 기존의 PNG, JPEG보다 더 작은 파일 크기를 제공
    - 브라우저 호환성을 위해 대체 이미지를 제공하자



⬅ iOS 16 미만 기기는 AVIF 미지원



⬅ 호환성이 더 좋은 WebP나 PNG를 대체 이미지로 삽입해둘 필요가 있음

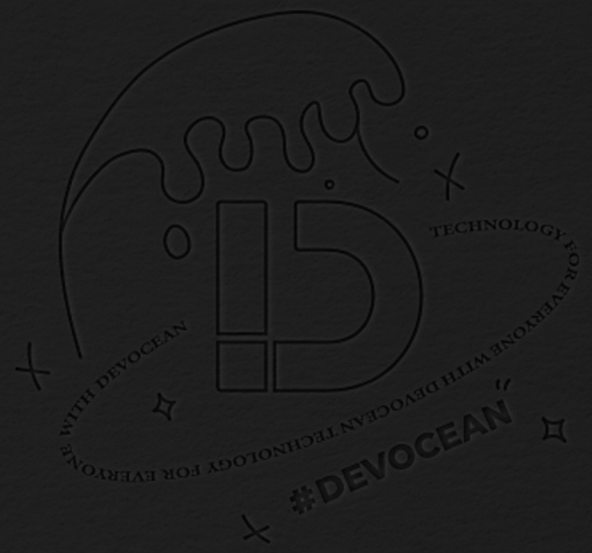
출처: https://caniuse.com/

03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
4. CDN(Content Delivery Network) 활용 - 어디서 이미지를 가져올 것인가?

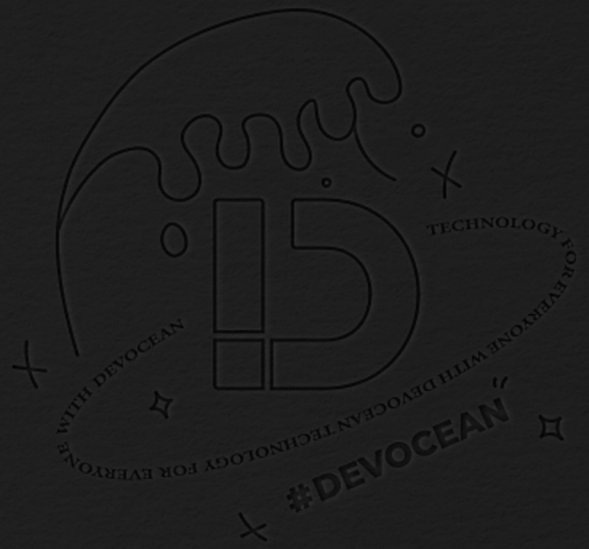
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
4. CDN(Content Delivery Network) 활용 - 어디서 이미지를 가져올 것인가?
    - 사용자와 가까운 서버에서 이미지를 제공할 수 있어 로딩 속도 향상



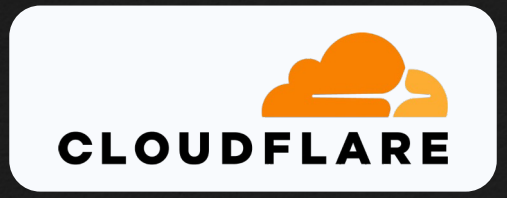
03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
4. CDN(Content Delivery Network) 활용 - 어디서 이미지를 가져올 것인가?
    - 사용자와 가까운 서버에서 이미지를 제공할 수 있어 로딩 속도 향상
    - 일부 CDN의 경우 URL parameter를 이용해 손쉬운 리사이징, 파일 형식 변경이 가능



03 이미지 최적화 - 기법 소개

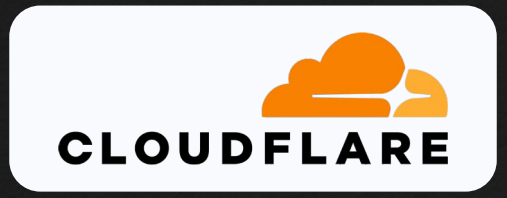
1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
4. CDN(Content Delivery Network) 활용 - 어디서 이미지를 가져올 것인가?
   - 사용자와 가까운 서버에서 이미지를 제공할 수 있어 로딩 속도 향상
   - 일부 CDN의 경우 URL parameter를 이용해 손쉬운 리사이징, 파일 형식 변경이 가능



https://<ZONE>/cdn-cgi/image/<OPTIONS>/<SOURCE-IMAGE>

03 이미지 최적화 - 기법 소개

1. 이미지 지연 로딩 - 언제 이미지를 로드할 것인가?
2. 이미지 리사이징 - 얼마나 크게 이미지를 보여줄 것인가?
3. 이미지 파일 형식 변경 - 어떤 형식으로 이미지를 저장할 것인가?
4. CDN(Content Delivery Network) 활용 - 어디서 이미지를 가져올 것인가?
   - 사용자와 가까운 서버에서 이미지를 제공할 수 있어 로딩 속도 향상
   - 일부 CDN의 경우 URL parameter를 이용해 손쉬운 리사이징, 파일 형식 변경이 가능

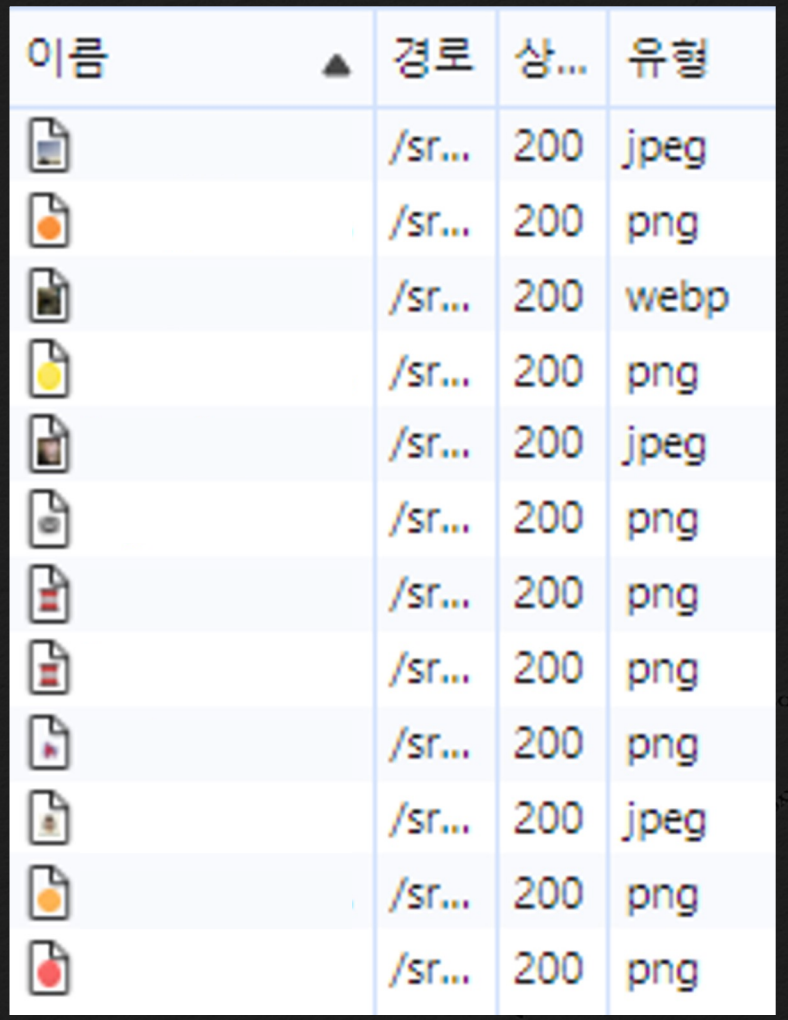


https://<ZONE>/cdn-cgi/image/width=250,height=250/<SOURCE-IMAGE>

출처: https://developers.cloudflare.com/images/transform-images/transform-via-url/

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

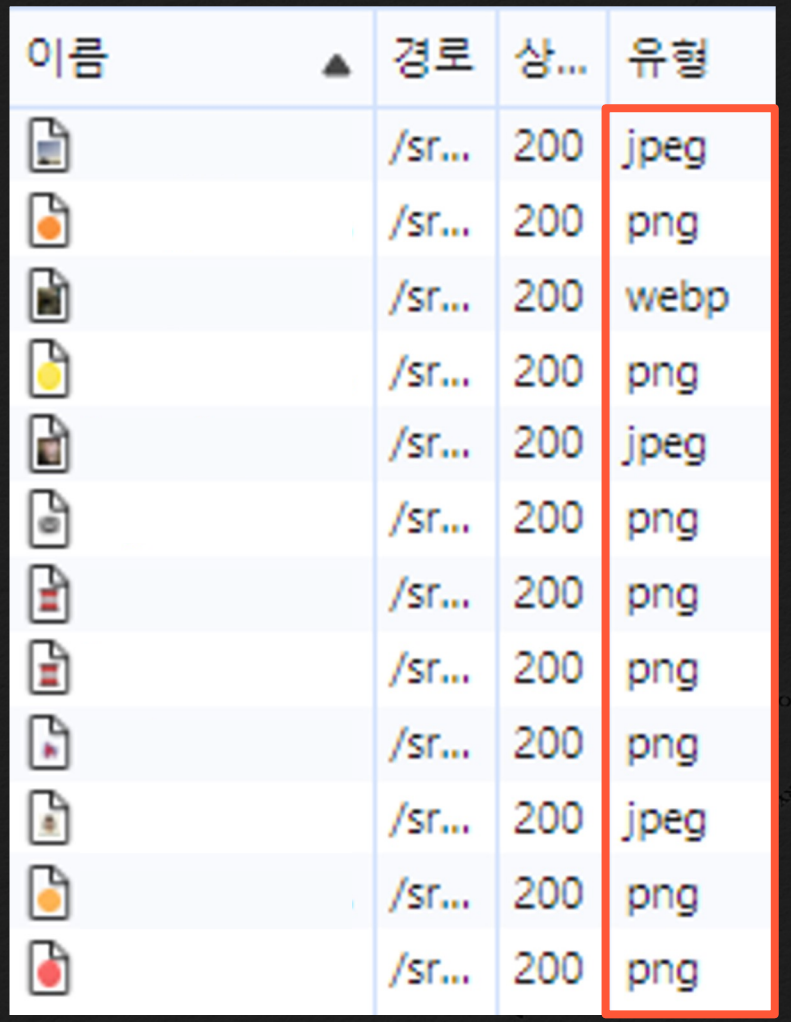
- 개선점
  - 한 번에 모든 이미지를 로드 중



이름	경로	상...	유형
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	webp
	/sr...	200	png
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	png
	/sr...	200	png
	/sr...	200	png
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	png

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 한 번에 모든 이미지를 로드 중
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG

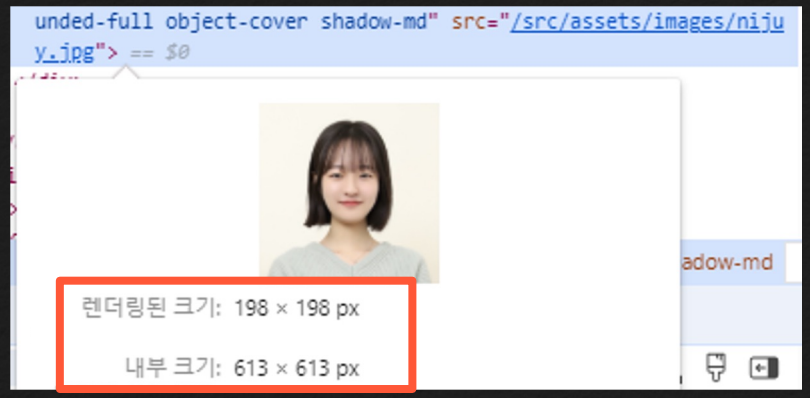


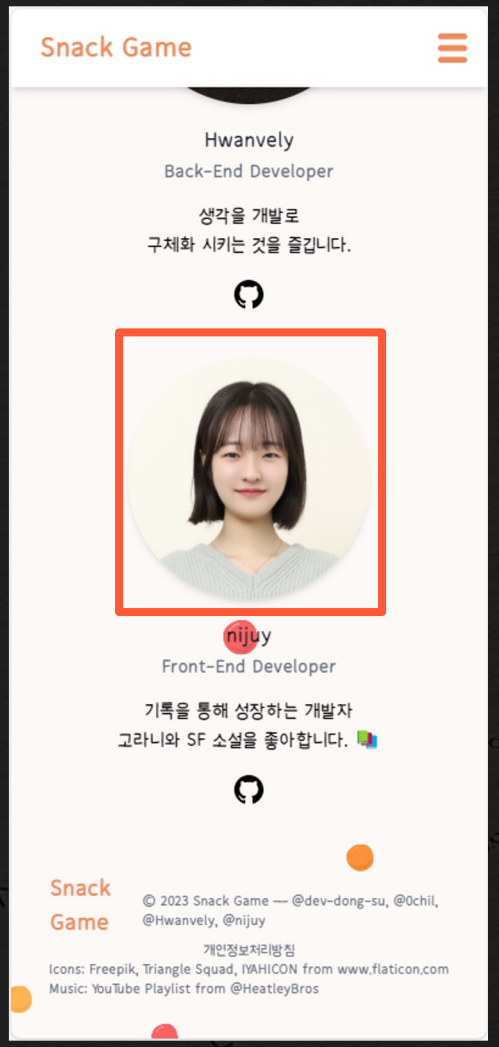
이름	경로	상...	유형
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	webp
	/sr...	200	png
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	png
	/sr...	200	png
	/sr...	200	png
	/sr...	200	jpeg
	/sr...	200	png
	/sr...	200	png

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 한 번에 모든 이미지를 로드 중
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG
  - 화면에서의 크기보다 과하게 큰 원본 파일의 크기

(198 * 198) (613 * 613)





03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

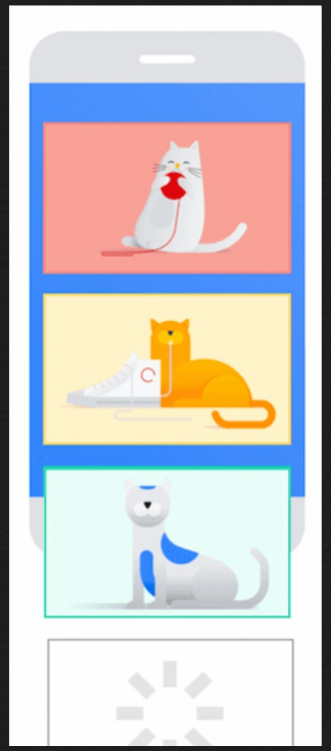
- 개선점
  - 한 번에 모든 이미지를 로드 중 ➡️ 페이지 하단에 있는 이미지에 loading="lazy"적용
  - 브라우저 레벨에서 지원하는 지연 로드 기법

```
<img
  loading="lazy"
  src={imgSrc}
  alt="team"
/>
```

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 한 번에 모든 이미지를 로드 중 ➡️ 페이지 하단에 있는 이미지에 loading="lazy"적용
  - 브라우저 레벨에서 지원하는 지연 로드 기법

```
<img
  loading="lazy"
  src={imgSrc}
  alt="team"
/>
```



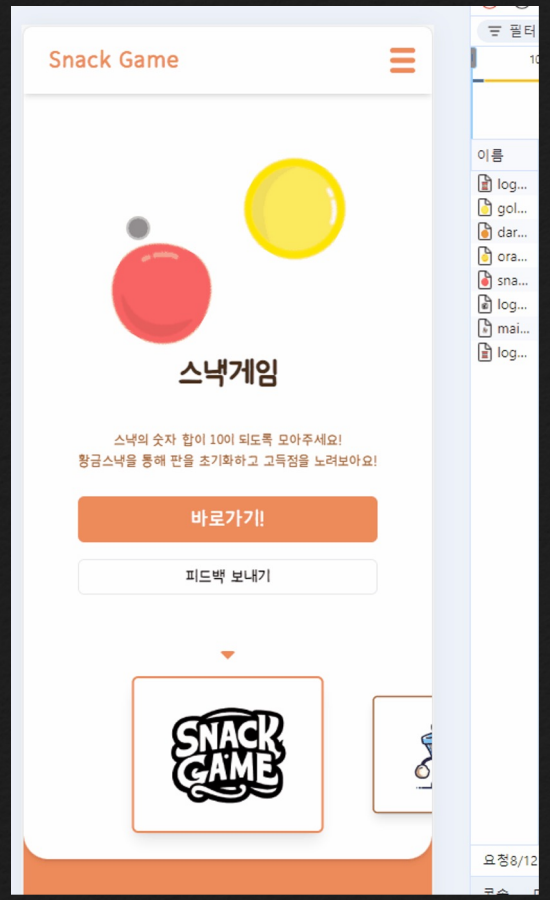
⬅️ 하단에 있는 이미지는 처음부터 로드하지 않음

출처: https://web.dev/articles/browser-level-image-lazy-loading

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 한 번에 모든 이미지를 로드 중 ➡️ 페이지 하단에 있는 이미지에 loading="lazy"적용
  - 브라우저 레벨에서 지원하는 지연 로드 기법

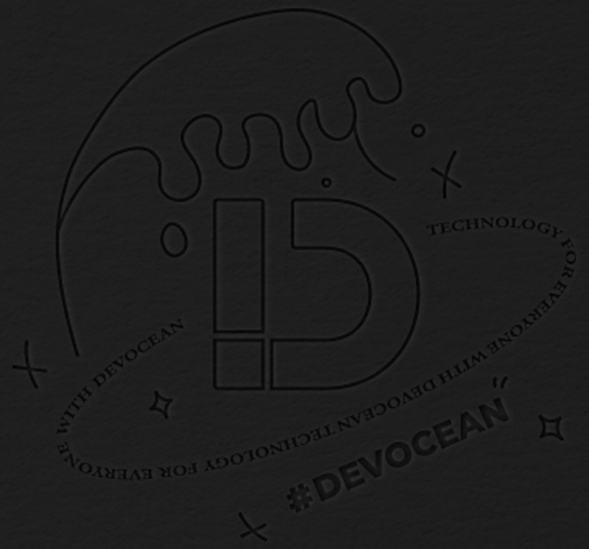
```
<img
  loading="lazy"
  src={imgSrc}
  alt="team"
/>
```



⬅️ 스크롤을 내리자
하단에 있는 이미지 로드가 시작되는 모습

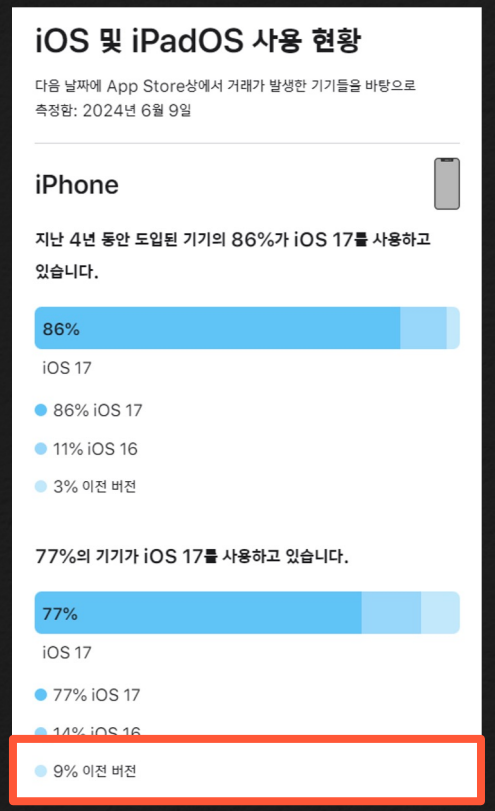
03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG ➡️ WebP/AVIF 형식으로 변환



03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG ➡️ WebP/AVIF 형식으로 변환
  - AVIF 채택 검토 과정에서의 사용자 환경 분석



+ Analytics

출처: https://developer.apple.com/kr/support/app-store/

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

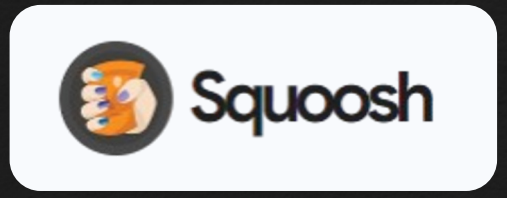
- 개선점
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG ➡️ WebP/AVIF 형식으로 변환
  - AVIF 채택 검토 과정에서의 사용자 환경 분석
  - 앱의 최소 설치 기준과 브라우저 접속 가능성을 고려하여 대체 이미지로 WebP 형식 채택

호환성
iPhone
iOS 13.4 이상 필요

```
<picture>
  <source srcSet={LogoImage} type="image/avif" />
  <img
    src={LogoWebPImage}
    alt={'blog image'}
  />
</picture>
```

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG ➡️ WebP/AVIF 형식으로 변환
  - 화면에서의 크기보다 과하게 큰 원본 파일의 크기 ➡️ 리사이징 진행



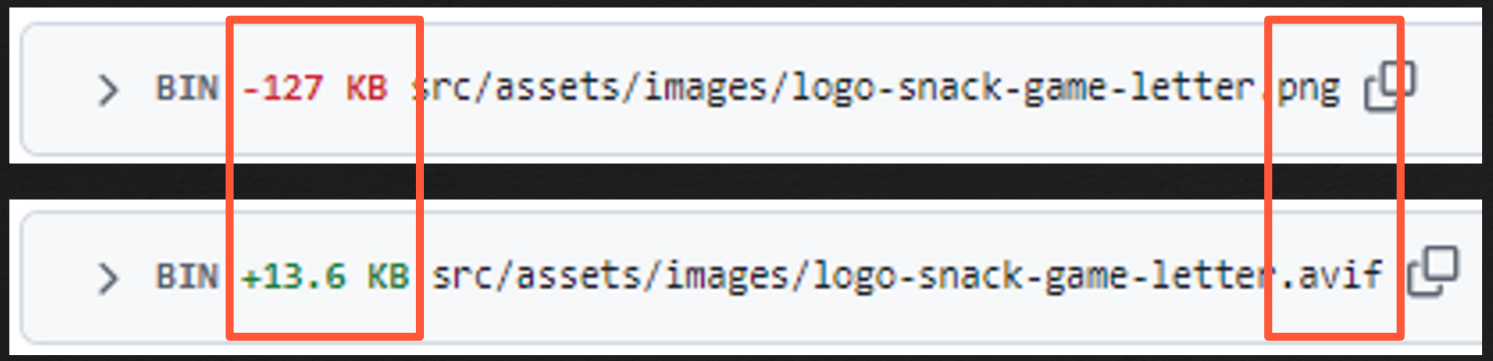
Google Chrome Labs에서 개발한 웹 기반 이미지 최적화 툴
  - 사이즈, 압축률, 화질 등 세부 설정 가능
  - 실시간 화질 비교 기능 제공

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 개선점
  - 대부분의 이미지가 압축률이 낮은 PNG/JPEG ➡️ WebP/AVIF 형식으로 변환
  - 화면에서의 크기보다 과하게 큰 원본 파일의 크기 ➡️ 리사이징 진행

Squoosh

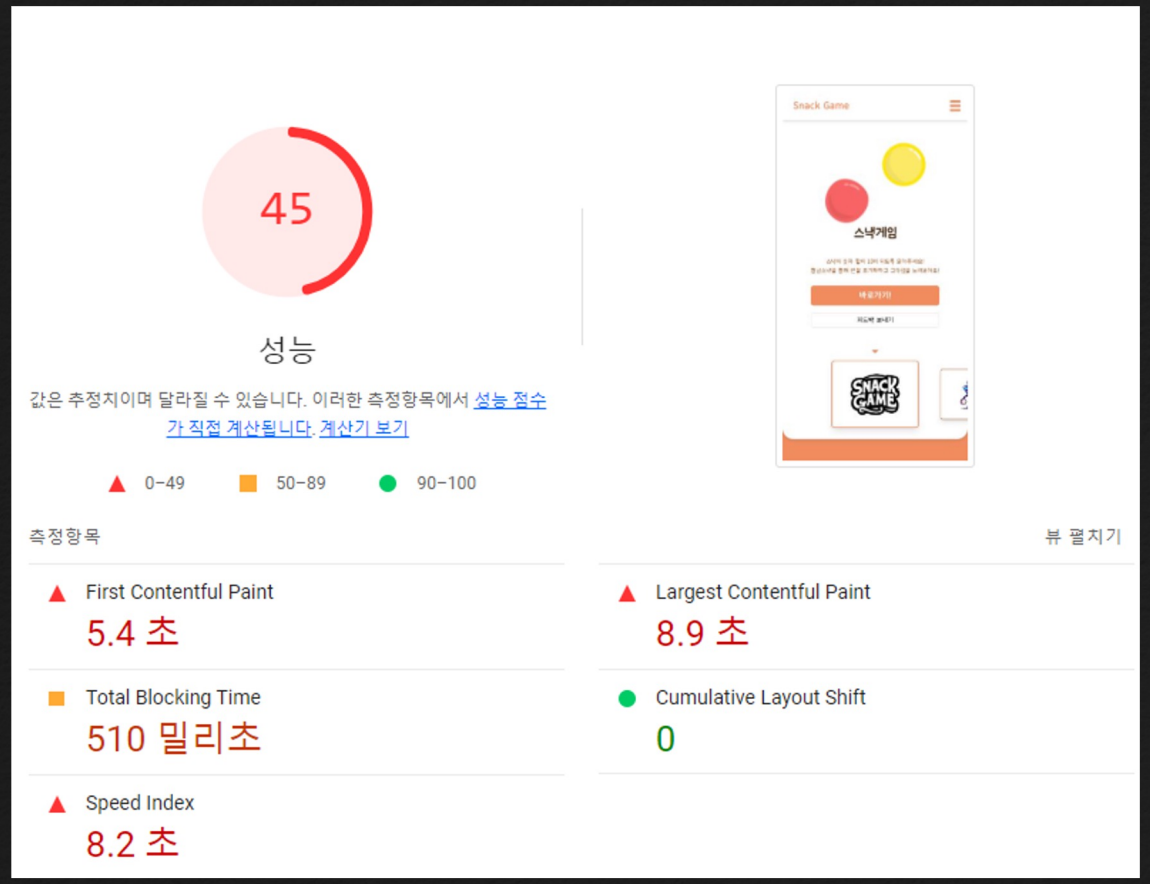
Google Chrome Labs에서 개발한 웹 기반 이미지 최적화 툴
  - 사이즈, 압축률, 화질 등 세부 설정 가능
  - 실시간 화질 비교 기능 제공

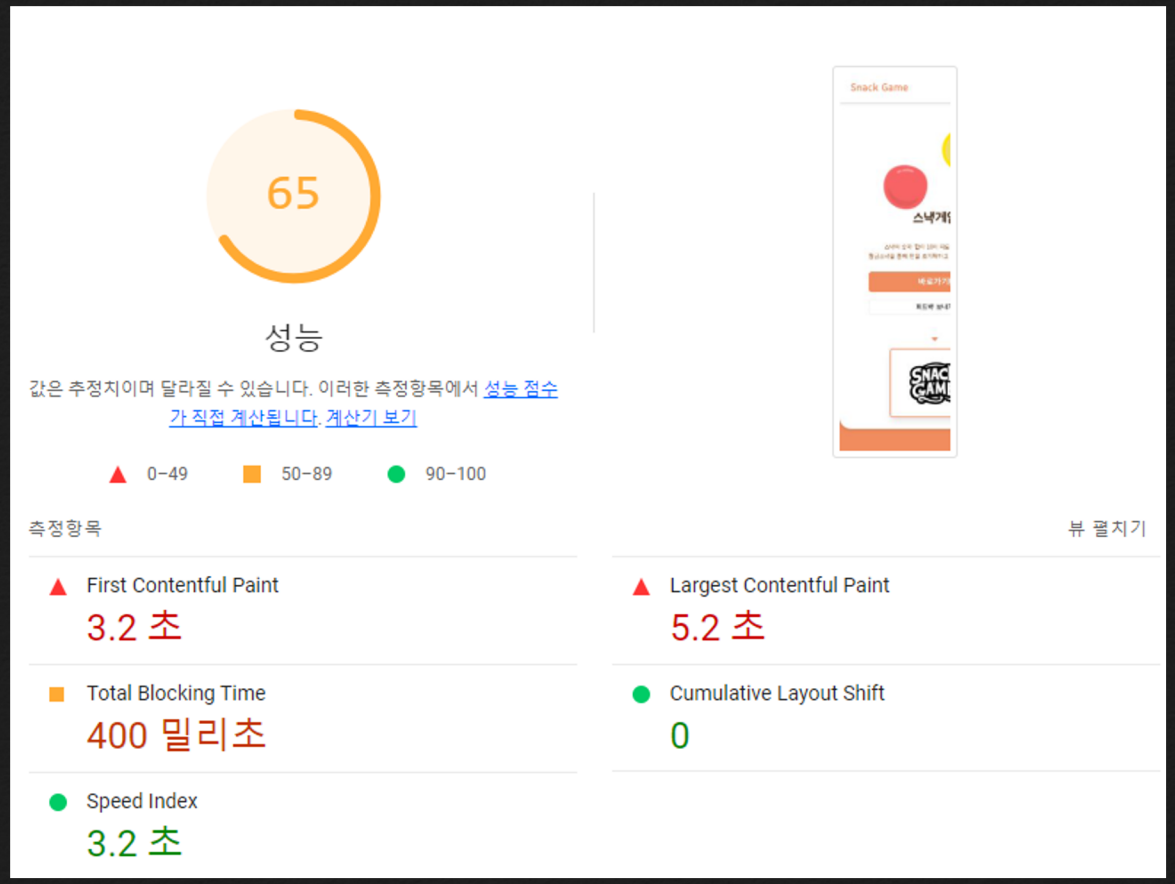


파일 용량 89% 감소

03 이미지 최적화 - 프로젝트 적용 사례: 스낵게임

- 랜딩 페이지 개선 후 점수
  - FCP 40% 감소
  - LCP 42% 감소

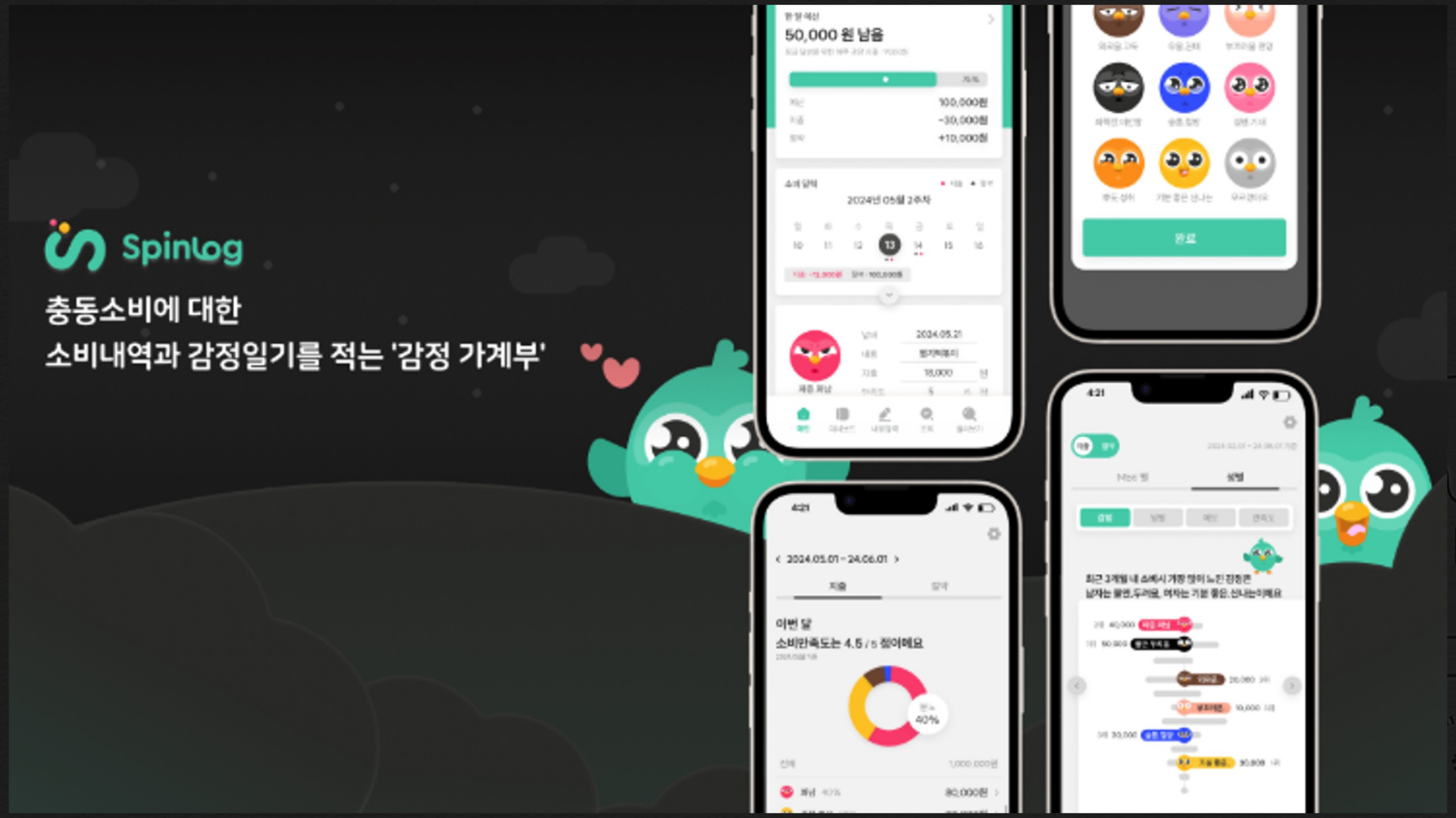




04 폰트 최적화 및 적용 사례

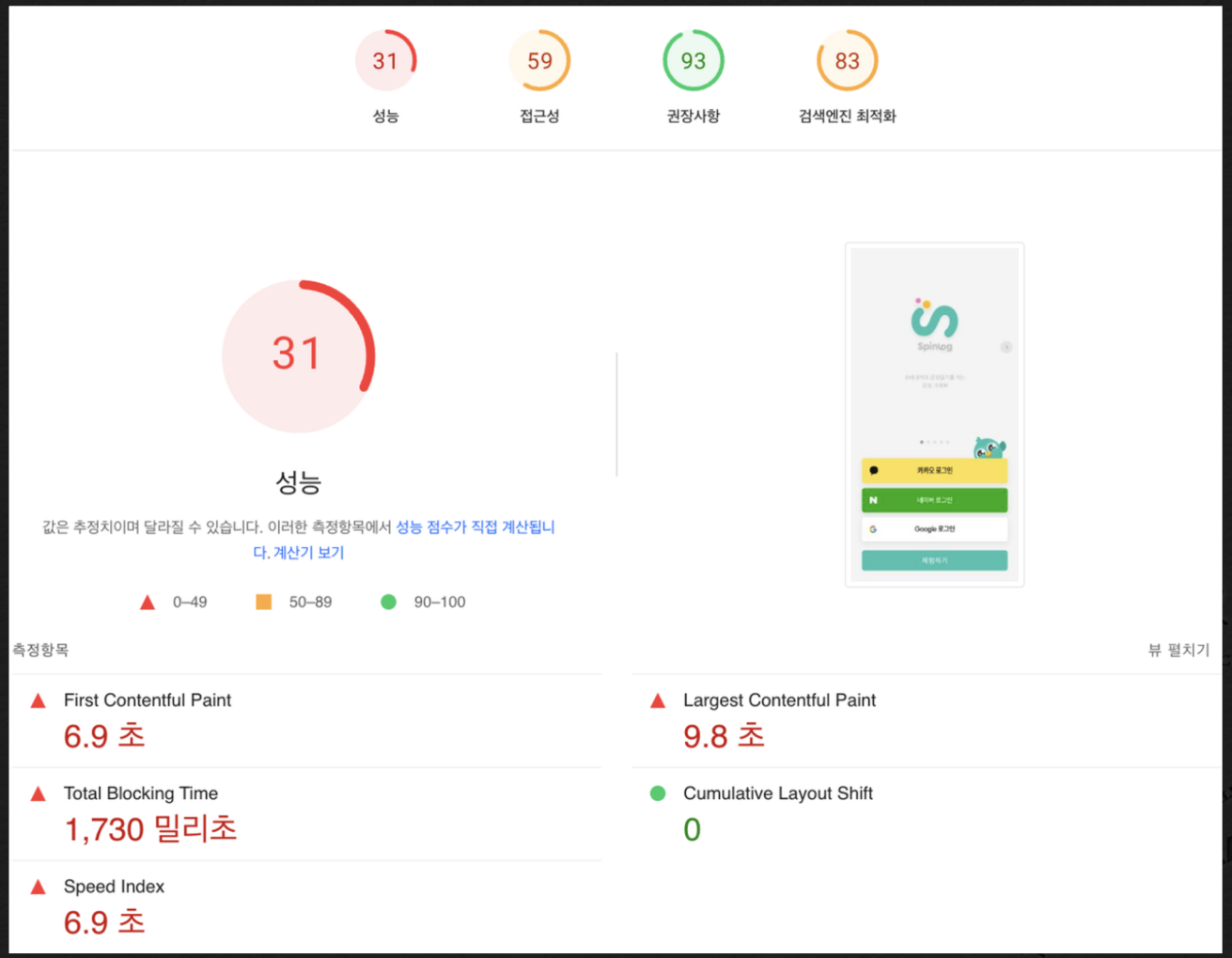
04 폰트 최적화 및 적용 사례

서비스 소개



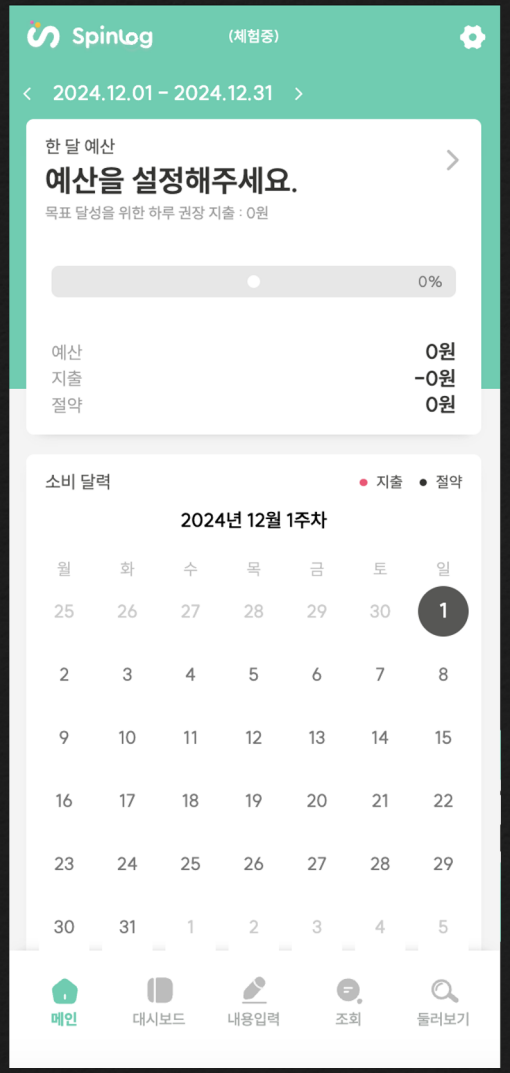
04 폰트 최적화 및 적용 사례

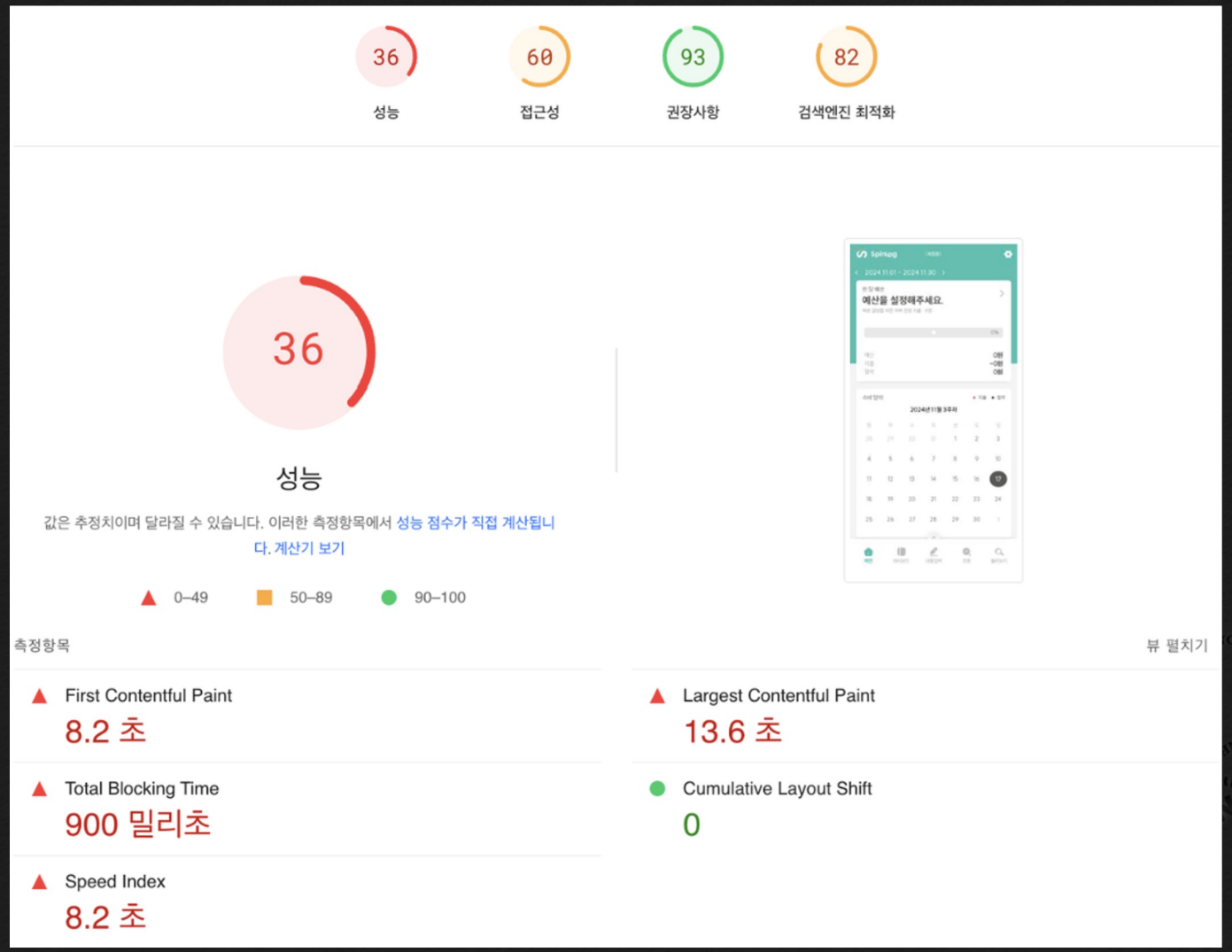
로그인 페이지



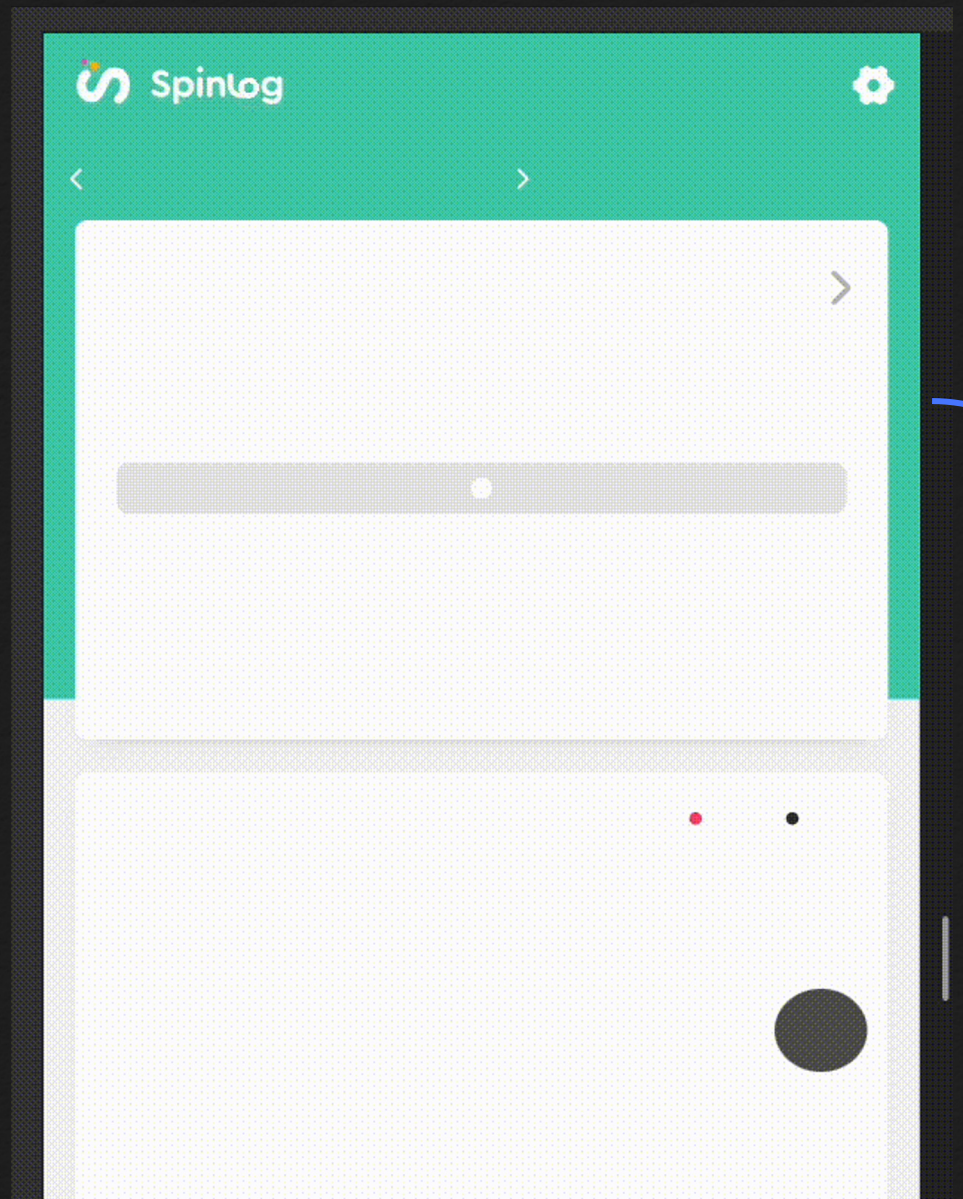
04 폰트 최적화 및 적용 사례

메인 페이지



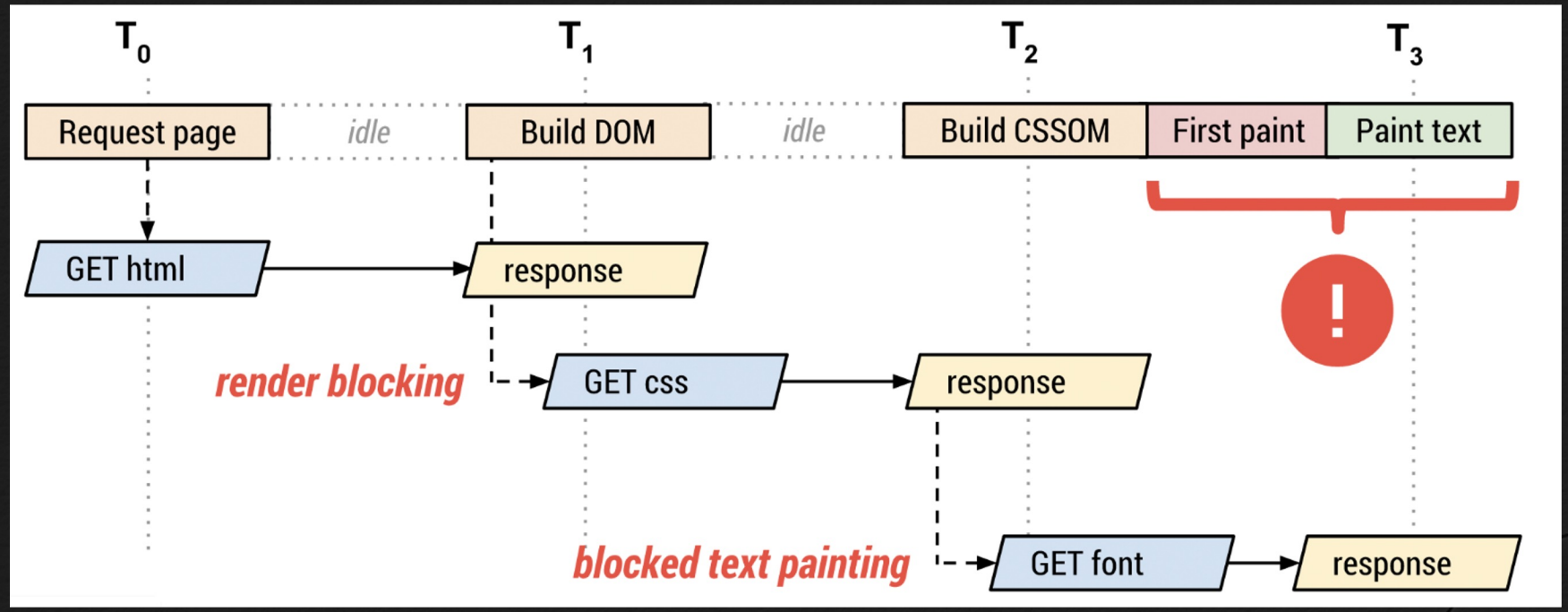


04 폰트 최적화 및 적용 사례



FOUT 현상 발생 !

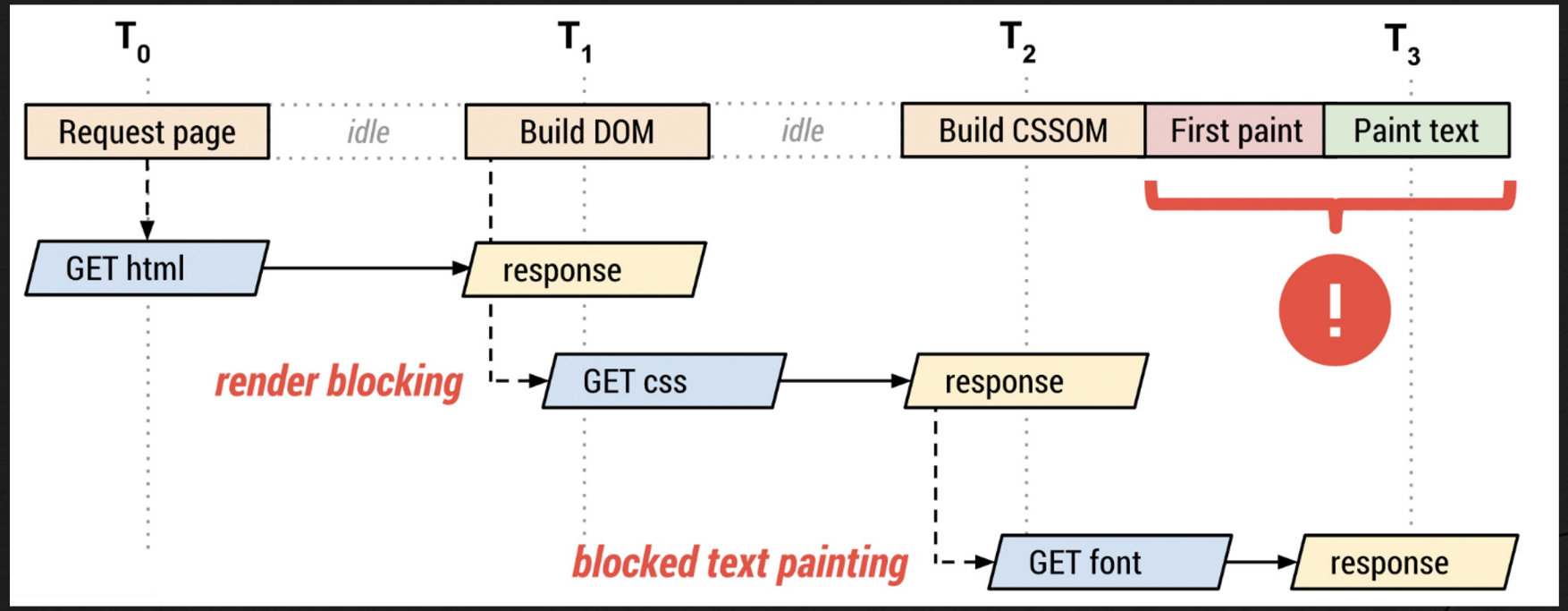
04 폰트 최적화 및 적용 사례 - 브라우저 렌더링 과정



출처 : web.dev

1. 브라우저가 HTML을 요청 -> 응답받은 HTML로 DOM 구성
2. 브라우저는 필요한 CSS 확인하여 요청 -> 응답받은 CSS로 CSSOM 구성
3. 브라우저는 렌더링에 필요한 폰트 요청
4. 브라우저는 폰트 요청의 응답을 기다리지 않고 렌더링 진행

04 폰트 최적화 및 적용 사례 - 브라우저 렌더링 과정



출처 : web.dev

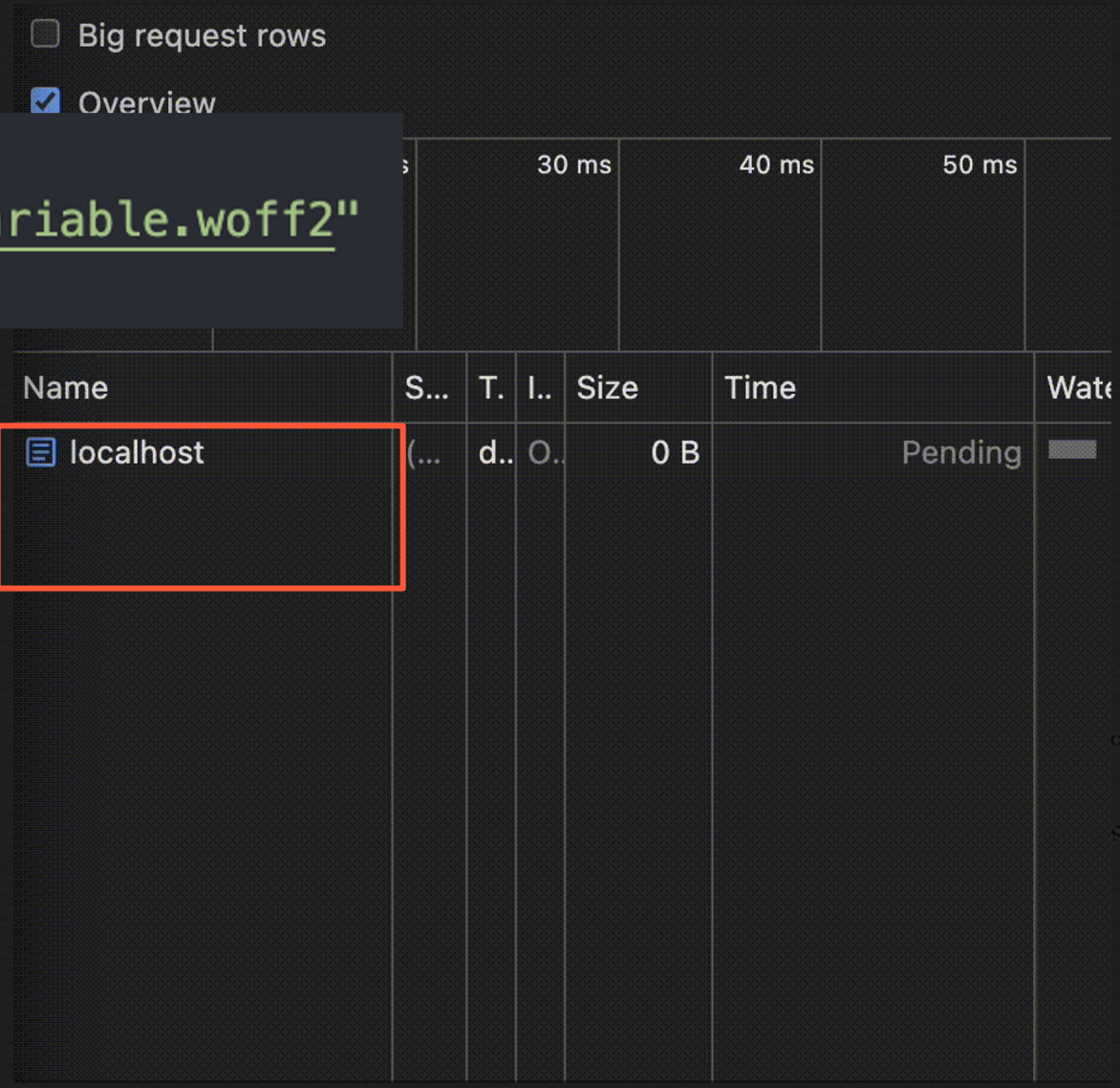
폰트 최적화 방법

1. 폰트 파일 사전 로드
2. font-display 설정
3. 폰트 파일 크기 축소

04 폰트 최적화 및 적용 사례 - 폰트 파일 사전 로딩

link 태그를 활용한 사전 로딩

```
<link rel="preload"
      href="/assets/fonts/SUIT/SUIT-Variable.woff2"
      as="font" type="font/woff2" />
```



폰트파일 로드가 우선적으로 요청된다. ⟶

04 폰트 최적화 및 적용 사례 - font-display

font-display 속성을 이용해 폰트가 적용되는 시점을 제어

auto	브라우저 기본 동작 (Edge: FOIT / Chrome, Safari, Firefox: FOUT)
block	FOIT (timeout = 3s)
swap	FOUT
fallback	FOIT (timeout = 0.1s) / 3초 후에도 폰트를 불러오지 못한 경우, 기본 폰트로 유지, 이후 캐시
optional	FOIT (timeout = 0.1s) / 이후 네트워크 상태에 따라 기본 폰트로 유지할지 결정, 이후 캐시

FOUT (Flash of Unstyled Text)

폰트의 다운로드 여부와 상관없이 먼저 텍스트를 보여준 후, 폰트가 다운로드되면 폰트를 적용하는 방식

ex) 중요한 텍스트

FOIT (Flash of Invisible Text)

폰트가 완전히 다운로드 되기 전까지는 텍스트 자체를 보여주지 않고, 폰트가 다운로드되면 폰트가 적용된 텍스트를 보여주는 방식

ex) 꼭 전달하지 않아도 되는 텍스트

fontfaceobserver같은 라이브러리를 활용해서 fade-in 애니메이션과 함께 활용 가능

04 폰트 최적화 및 적용 사례 - 폰트 파일 크기 줄이기

폰트 파일의 크기를 줄이면, 폰트가 다운로드 되는 시간을 단축할 수 있다.

EOT > TTF/OFF > WOFF > WOFF2

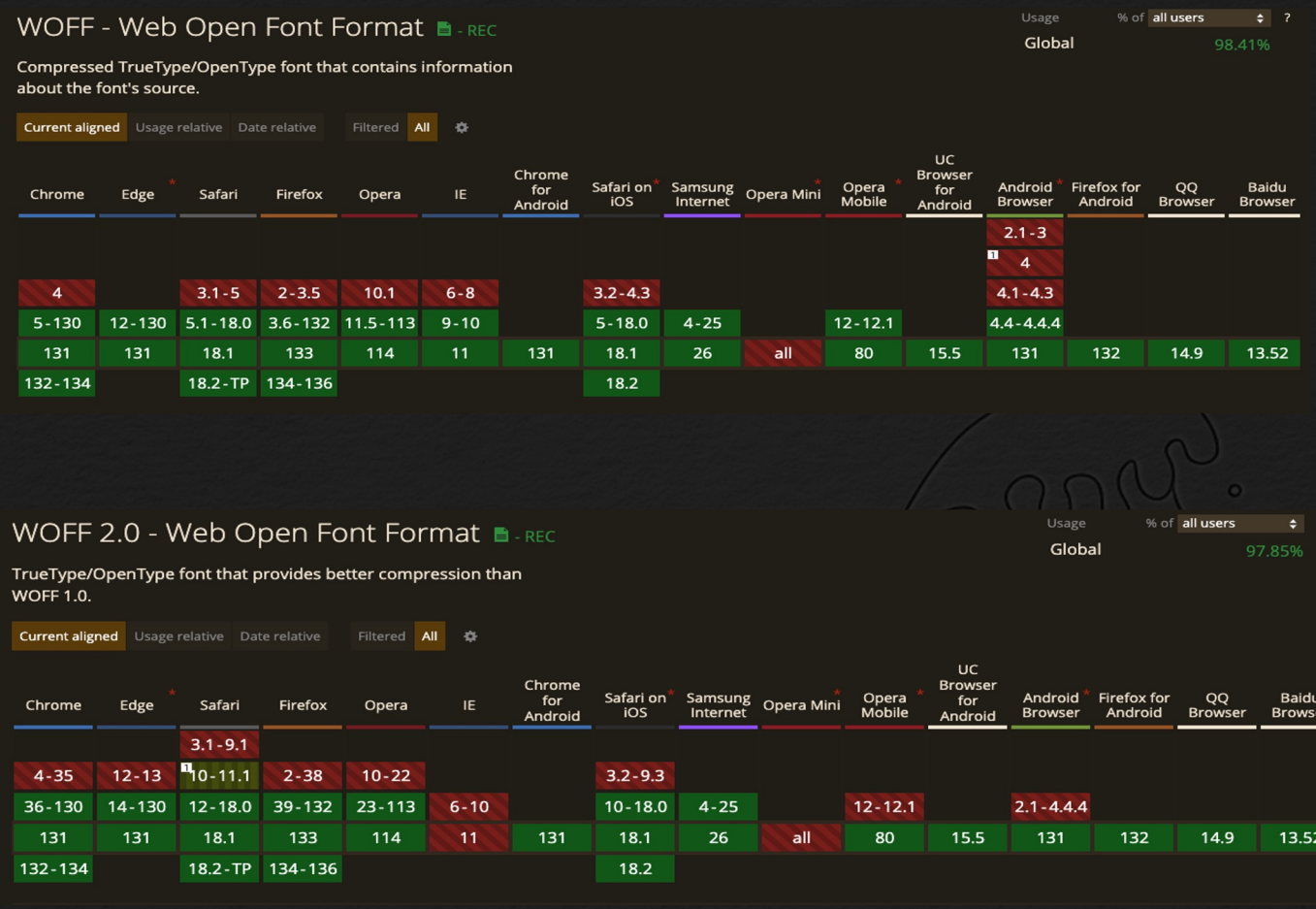
폰트 포맷별 파일 크기 비교

04 폰트 최적화 및 적용 사례 - 폰트 파일 크기 줄이기

브라우저 호환성도 고려해야 합니다.

WOFF, WOFF2

(Web Open Font Format)



WOFF - Web Open Font Format - REC

Compressed TrueType/OpenType font that contains information about the font's source.

Usage: Global — % of all users: 98.41%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser
												2.1-3			
												4			
4		3.1-5	2-3.5	10.1	6-8		3.2-4.3					4.1-4.3			
5-130	12-130	5.1-18.0	3.6-132	11.5-113	9-10		5-18.0	4-25		12-12.1		4.4-4.4.4			
131	131	18.1	133	114	11	131	18.1	26	all	80	15.5	131	132	14.9	13.52
132-134		18.2-TP	134-136				18.2								

WOFF 2.0 - Web Open Font Format - REC

TrueType/OpenType font that provides better compression than WOFF 1.0.

Usage: Global — % of all users: 97.85%

Current aligned | Usage relative | Date relative | Filtered | All

Chrome	Edge	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS	Samsung Internet	Opera Mini	Opera Mobile	UC Browser for Android	Android Browser	Firefox for Android	QQ Browser	Baidu Browser
		3.1-9.1													
4-35	12-13	10-11.1	2-38	10-22			3.2-9.3								
36-130	14-130	12-18.0	39-132	23-113	6-10		10-18.0	4-25		12-12.1		2.1-4.4.4			
131	131	18.1	133	114	11	131	18.1	26	all	80	15.5	131	132	14.9	13.52
132-134		18.2-TP	134-136				18.2								

출처 : can i use

04 폰트 최적화 및 적용 사례 - 폰트 최적화 방법: 폰트 파일 크기 줄이기

폰트 포맷 변경하기

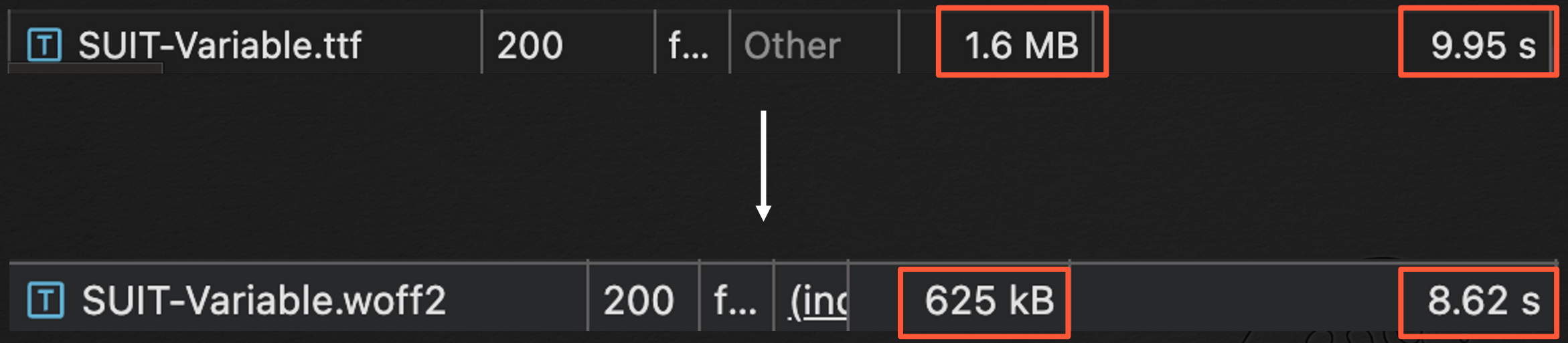
```
import { createGlobalStyle } from 'styled-components';

const GlobalFonts = createGlobalStyle`
  @font-face {
    font-family: 'SUIT';
    font-weight: 100 900;
    font-display: swap;
    src: url('/assets/fonts/SUIT/SUIT-Variable.woff2') format('woff2-variations'),
         url('/assets/fonts/SUIT/SUIT-Variable.ttf') format('truetype');
  }
`;

export default GlobalFonts;
```

04 폰트 최적화 및 적용 사례 - 폰트 최적화 방법: 폰트 파일 크기 줄이기

폰트 파일 다운로드 크기 및 속도 비교 (Network throttling Slow 4G 기준)



크기	61% 감소
시간	13.4% 단축

04 폰트 최적화 및 적용 사례 - 폰트 파일 크기 줄이기: 서브셋 폰트, Data-URI

서브셋 폰트 사용하기

웹 폰트 파일에서 필요한 문자 집합만을 선택하여 추출한 폰트 파일

Transfonter와 같은 폰트 지원 서비스나, 다양한 서브셋 폰트 생성기 사용

-> 필요한 일부 문자의 폰트 정보만 가지고 있기 때문에 파일 크기가 작아짐

Data-URI 사용하기 **Data-URI: data 스킴이 접두어로 붙은 문자열 형태의 데이터*

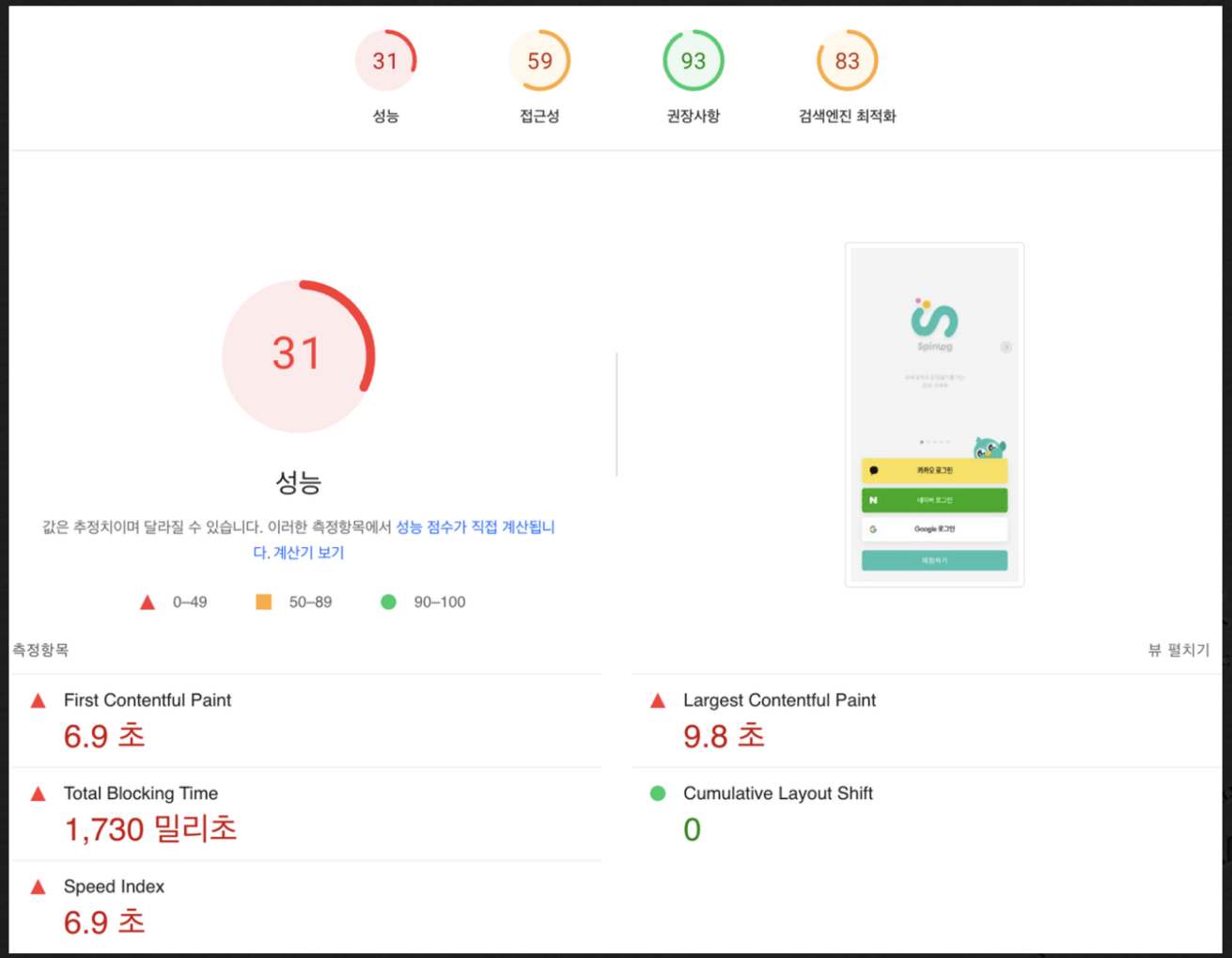
폰트를 파일 형태가 아닌 Data-URI 형태로 만들어 CSS 파일에 포함

Transfonter와 같은 폰트 지원 서비스에서 변환 가능

-> 별도의 네트워크 로드 없이 폰트를 사용, 단 Data-URI가 포함된 파일의 다운로드가 느려질 수 있으므로 주의 필요

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

로그인 페이지



04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

로그인 페이지

개선점 1.

바로 보이지 않는 스와이프 이미지에 지연 로딩 적용하기

이미지 사이즈 지정하기, 포맷 변경하기

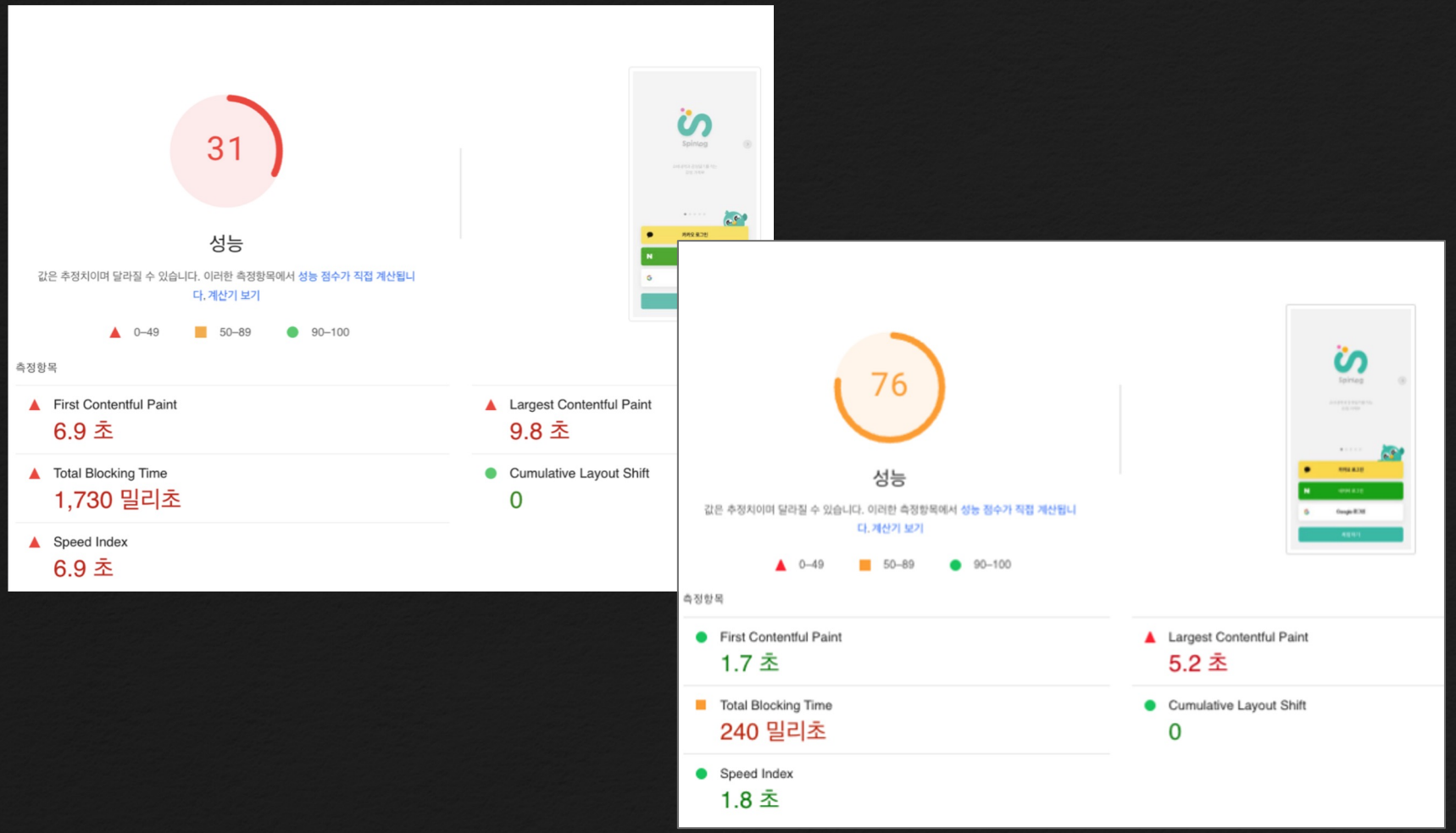
개선점 2.

바로 보여야 하는 이미지들에 사전 로딩 적용하기

이미지 사이즈 지정하기, 포맷 변경하기

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

로그인 페이지 최적화 전/후 비교

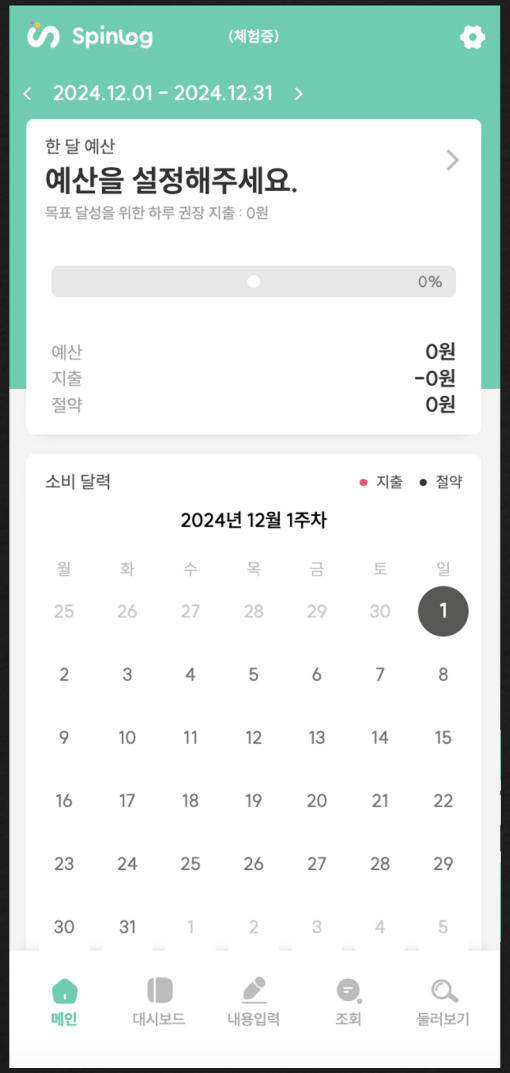


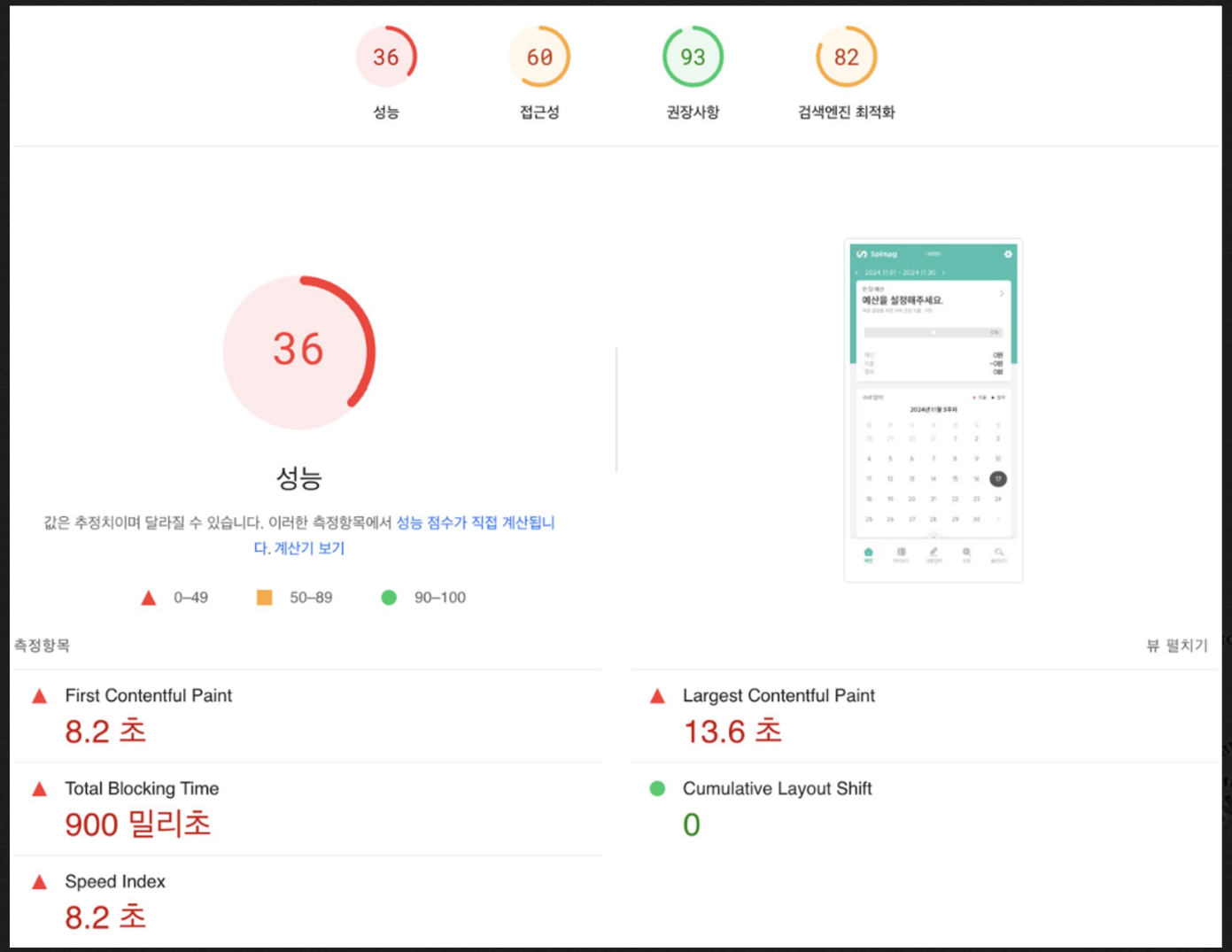
성능 점수
145.2% 개선

FCP	75.4% 감소
LCP	46.9% 감소
TBT	86.1% 감소
SI	73.9% 감소

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

메인 페이지





04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

메인 페이지

개선점 1.

페이지 컴포넌트 코드 분할하기,

바로 표시되지 않는 컴포넌트에 지연 로딩 적용하기

개선점 2.

script에 defer 속성 적용하기

개선점 3.

사용되지 않는 CSS 속성 정리 및 폰트 최적화 하기

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

routes.tsx

```tsx
import App from '../App';
import Layout from '../components/layout/Layout';
import ProtectedRoute from './protectedRoute/ProtectedRoute';
import MainPage from '../pages/main/MainPage';
import LoginPage from '../pages/login/LoginPage';
import ErrorPage from '../pages/error/ErrorPage';
import AuthPage from '../pages/auth/AuthPage';
import Loading from '@components/infor
import ExpenseListViewPage from '../pa
import DashboardPage from '../pages/da
import AddExpensePage from '../pages/a
import ExpenseDetailViewPage from '../
import StatisticsPage from '../pages/s

import SettingPage from '../pages/sett
```

```
dist/registerSW.js                  0.13 kB
dist/manifest.webmanifest           0.15 kB
dist/index.html                     3.71 kB | gzip:   1.63 kB
dist/assets/kakao-B_HqQA2J.png      7.17 kB
dist/assets/google-DgX5deSd.png     9.62 kB
dist/assets/intro2-BAVOm4Xn.webp   15.09 kB
dist/assets/naver-BQORu5_k.png     19.98 kB
dist/assets/intro1-BsPajrQH.webp   21.71 kB
dist/assets/intro4-q9mi0g7q.webp   21.98 kB
dist/assets/intro3-CDly0jwx.webp   23.52 kB
dist/assets/main-BwVu8tOa.css      14.99 kB | gzip:   4.40 kB
dist/assets/sw-CzuRWWfs.js          0.26 kB | gzip:   0.15 kB
dist/assets/main-D-FBTX1e.js    1,516.11 kB | gzip: 460.47 kB
```

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

코드분할 후

```
const MainPage = lazyWithRetries(() => import('../pages/main/MainPage'));

const AuthPage = lazyWithRetries(() => import('../pages/auth/AuthPage'));
const ErrorPage = lazyWithRetries(() => import('../pages/error/ErrorPage'));

const ExpenseListViewPage = lazyWithRetr
  () => import('../pages/expenseListView/
);
const DashboardPage = lazyWithRetries(()
const AddExpensePage = lazyWithRetries((
const ExpenseDetailViewPage = lazyWithRe
  () => import('../pages/expenseDetailVi
);
const StatisticsPage = lazyWithRetries((
const SettingPage = lazyWithRetries(() =>
```

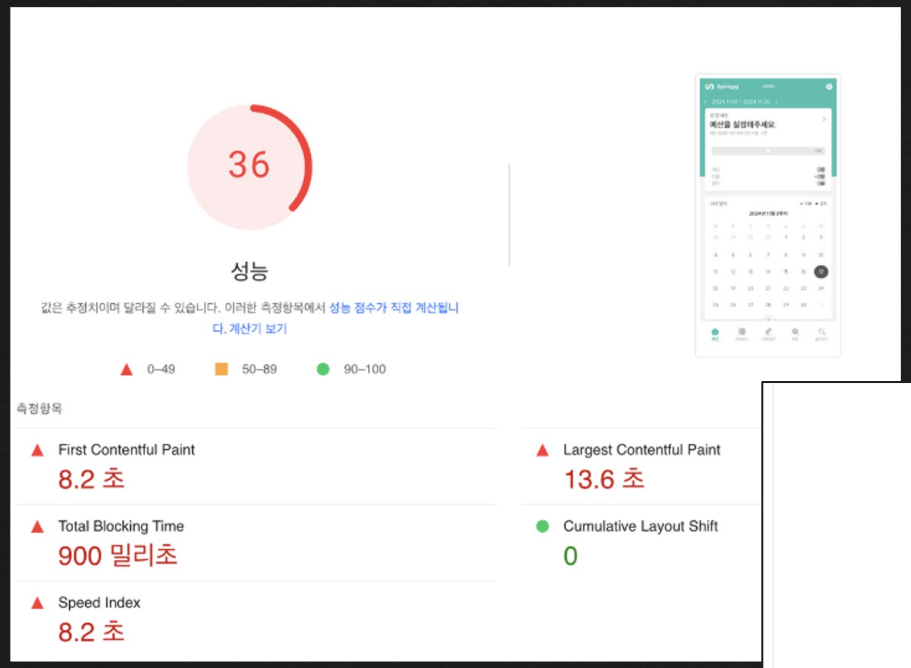
```
dist/assets/Calendar-CJqrBvut.js                5.97 kB | gzip:   2.63 kB
dist/assets/BottomNavigation-DwSgYZx6.js        6.39 kB | gzip:   2.18 kB
dist/assets/SettingPage-BCEgTuWJ.js             7.07 kB | gzip:   2.81 kB
dist/assets/AuthPage-CRLVq1MV.js                8.01 kB | gzip:   3.03 kB
dist/assets/DashboardPage-BoHfYuLl.js           8.50 kB | gzip:   3.28 kB
dist/assets/useBaseQuery-BclIltNi.js           10.22 kB | gzip:   3.15 kB
dist/assets/ExpenseDetailViewPage-Cqp22741.js  10.41 kB | gzip:   3.94 kB
dist/assets/ErrorPage-7udJrCPp.js              11.36 kB | gzip:   3.85 kB
dist/assets/AddExpensePage-BOSQmqJo.js         12.74 kB | gzip:   5.15 kB
dist/assets/ExpenseListViewPage-BuECfw8j.js    15.67 kB | gzip:   5.22 kB
dist/assets/index.esm-BO41jno-.js              23.64 kB | gzip:   8.98 kB
dist/assets/index-D4TwTRvq.js                  47.35 kB | gzip:  11.58 kB
dist/assets/axios-b_nLmLIg.js                  56.23 kB | gzip:  20.83 kB
dist/assets/lodash-COjyWtB6.js                 71.48 kB | gzip:  26.46 kB
dist/assets/StatisticsPage-tvqM6mCY.js        158.05 kB | gzip:  56.42 kB
dist/assets/main-CMJyg5MW.js                  496.26 kB | gzip: 156.67 kB
dist/assets/react-apexcharts.min-DJcZmr9p.js  538.26 kB | gzip: 142.61 kB
```

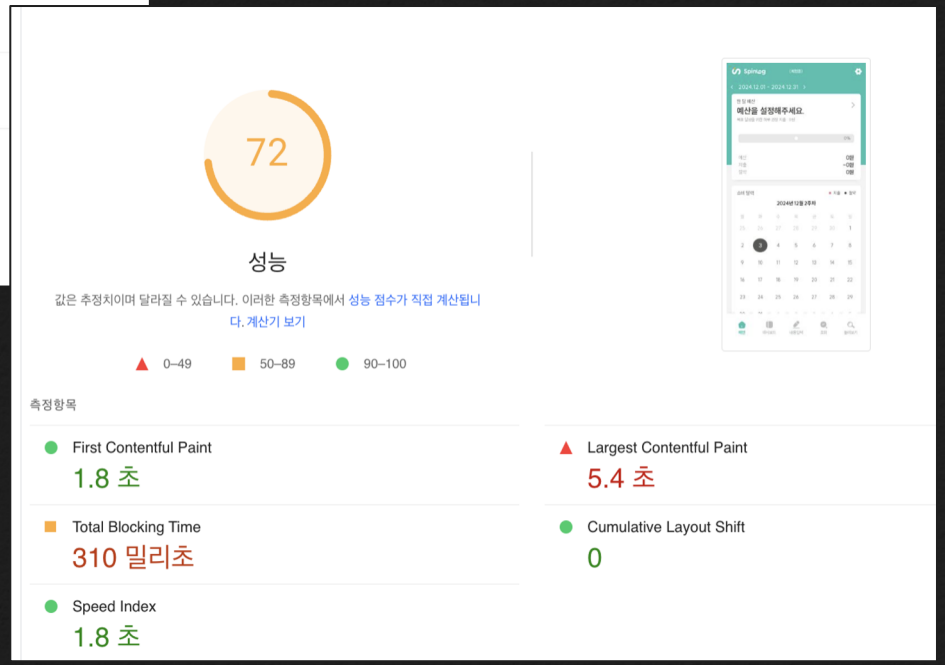
가장 큰 파일이

67.3% 감소

04 폰트 최적화 및 적용 사례 - 프로젝트 적용 사례: SpinLog

메인 페이지 최적화 전/후 비교





성능 점수
100% 개선

FCP	78.0% 감소
LCP	60.3% 감소
TBT	65.6% 감소
SI	89.0% 감소

05 결론

05 결론

코드에 기름칠 하는 법

동영상 최적화

지연 로딩 & 사전 로딩

애니메이션 최적화

렌더링 최적화

레이아웃 이동 방지

이미지 최적화

Intersection Observer

불필요한 CSS 제거

병목 코드 최적화

텍스트 압축

폰트 최적화

코드 분할

캐시 최적화

Q&A

LinkedIn

이유진

임수현

장세영