

Task - Next.js 14 Remote Image Optimization and Accessible Avatar Component with Tailored Loading Strategies

Scenario & Objectives

Your project needs a **robust and performant** user profile component to display **user avatars**. These avatars may come from:

- Local assets (e.g., static images in `/public`)
- Remote CDN services like:
 - `example-bucket.s3.amazonaws.com/images/**`
 - `cdn.myapp.com/**`

The component must follow **Lighthouse best practices**, ensuring:

- Fast loading and responsive rendering
- Accessibility and SEO friendliness
- Secure handling of image sources
- Modern image optimizations via `next/image`

Tasks

1. Next.js Configuration

Update `next.config.js` with a **remotePatterns-based image configuration** that:

- Allows loading images from:
 - `example-bucket.s3.amazonaws.com/images/**`
 - `cdn.myapp.com/**`
- Uses the `images.remotePatterns` API, **not** `images.domains`
- Explain why `remotePatterns` is preferred in modern Next.js

[✓] *Your configuration must enable secure, scoped remote image support while allowing performance benefits from image optimization.*

2. React Component Implementation

Create a **TypeScript** React component named `Avatar.tsx` that:

Functional Requirements:

- Accepts the following props:

```
type AvatarProps = {
  src?: string;           // Optional image URL
  alt: string;            // Accessibility alt text (required)
  priority?: boolean;     // Whether to preload image (optional)
  size?: number;         // Optional size in pixels, default to 64
}
```

- Uses a **default local image** if `src` is not provided:

```
import defaultAvatar from '../public/avatar.png';
```

- Always sets:
 - `alt` for accessibility
 - `width / height` explicitly to avoid layout shift
 - `priority` only when necessary (e.g., profile image above the fold)
 - A responsive placeholder (`blurDataURL`) for better perceived performance

Optimization Notes (Include in Comments):

- Explain how `<Image>` from `next/image` :
 - Automatically **resizes** and **converts** images to modern formats (e.g., WebP/AVIF)
 - Supports **lazy loading** and **blur-up placeholders** for smooth UX
 - Prevents **Cumulative Layout Shift** (CLS) by enforcing width/height
 - Behaves differently in **development vs. production** (e.g., full-size fallback in dev)

Bonus (if time allows):

- Ensure the image is **visually clipped to a circle** using Tailwind CSS or inline styles (e.g., `rounded-full` , `object-cover`)
- Add support for `loading="eager"` or `loading="lazy"` based on `priority` prop

Constraints

- Do not use plain `` tags.
- Do not allow arbitrary domains unless configured.
- Do not skip `alt` attributes.
- Do not use `layout="fill"` unless in a wrapper with `position-relative`.
- Do not hardcode inline styles for accessibility-critical attributes.

Deliverables:

- `next.config.js` snippet
- `Avatar.tsx` component code
- Inline comments describing performance and accessibility considerations

Paramount Ventures