

# Chery Bangladesh Design System Guidelines

## Brand Overview

The Chery Bangladesh design system creates a premium automotive digital experience that balances modern aesthetics with functional clarity. The design language emphasizes clean lines, subtle animations, and a minimalist approach that puts content and vehicle imagery at the forefront.

## Core Design Principles

### 1. Elegant Minimalism

- Favor clean, uncluttered layouts with ample white space
- Use subtle borders and separators to create visual hierarchy
- Emphasize content through typography and spacing rather than excessive decorative elements

### 2. Purposeful Motion

- Animations should enhance usability, not distract
- Use motion to guide attention and provide feedback
- Apply consistent timing and easing for cohesive experience

### 3. Content Focus



- Vehicle imagery and key messaging should be prominent
- Use gradient overlays to maintain text legibility over images
- Create clear visual hierarchy that guides users through the content

### 4. Premium Feel

- Attention to detail in spacing, alignment, and typography
- Subtle interactions that feel refined and intentional
- Consistent components that work together harmoniously

## Color Palette

### Primary Colors

- **Primary Red:**  #D10000 (Tailwind variable: `primary`)
- **Primary Dark:**  #A60000 (Tailwind variable: `primary-dark`)

- **Black:** ☒ #000000
- **White:** ☐ #FFFFFF

## Neutral Palette

- **Gray 900:** ☒ #111827 (Used for headings and primary text)
- **Gray 600:** ☒ #4B5563 (Used for body text)
- **Gray 500:** ☒ #6B7280 (Used for secondary text)
- **Gray 200:** ☐ #E5E7EB (Used for borders)
- **Gray 100:** ☐ #F3F4F6 (Used for background accents)

## Supporting Colors

- **Success Green:** ☒ #10B981 (For success states)
- **Warning Yellow:** ☐ #F59E0B (For warning states)
- **Error Red:** ☐ #EF4444 (For error states)

## Typography

### Font Family

- System font stack with fallbacks:

CSS

```
font-family: ui-sans-serif, system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", Robot
```

### Font Sizes

- **Headings:**
  - H1:  (2.25rem to 3.75rem)
  - H2:  (1.875rem to 3rem)
  - H3:  (1.5rem to 1.875rem)
  - H4:  (1.25rem to 1.5rem)
  - H5:  (1.125rem to 1.25rem)
- **Body:**
  - Body Large:  (1.125rem)
  - Body:  (1rem)

- Body Small: `text-sm` (0.875rem)
- Caption: `text-xs` (0.75rem)

## Font Weights

- **Bold:** `font-bold` (700)
- **Medium:** `font-medium` (500)
- **Regular:** `font-normal` (400)

## Line Heights

- Headings: `leading-tight` (1.25)
- Body: `leading-normal` (1.5)
- Tight text: `leading-snug` (1.375)

## Spacing System

Follow the Tailwind CSS spacing scale:

- **Micro Spacing:** `space-x-1 space-y-1` (0.25rem)
- **Small Spacing:** `space-x-2 space-y-2` (0.5rem), `space-x-3 space-y-3` (0.75rem)
- **Medium Spacing:** `space-x-4 space-y-4` (1rem), `space-x-6 space-y-6` (1.5rem)
- **Large Spacing:** `space-x-8 space-y-8` (2rem), `space-x-12 space-y-12` (3rem)
- **Extra Large Spacing:** `space-x-16 space-y-16` (4rem), `space-x-20 space-y-20` (5rem)

## Layout

### Container

- Use `max-w-7xl mx-auto px-4 md:px-8` for consistent page width
- Ensure responsive padding adapts to screen sizes

### Grid System

- Use CSS Grid for complex layouts: `grid grid-cols-1 md:grid-cols-3 gap-6 md:gap-8`
- Use Flexbox for simpler alignment needs: `flex items-center justify-between`
- Common grid patterns:
  - 1-column mobile, 2-3 columns tablet, 4-12 columns desktop

## Section Spacing

- Vertical rhythm: `py-20` (5rem) for standard section padding
- Smaller sections: `py-12` (3rem) or `py-16` (4rem)
- Footer: `pt-16 pb-8` (top: 4rem, bottom: 2rem)

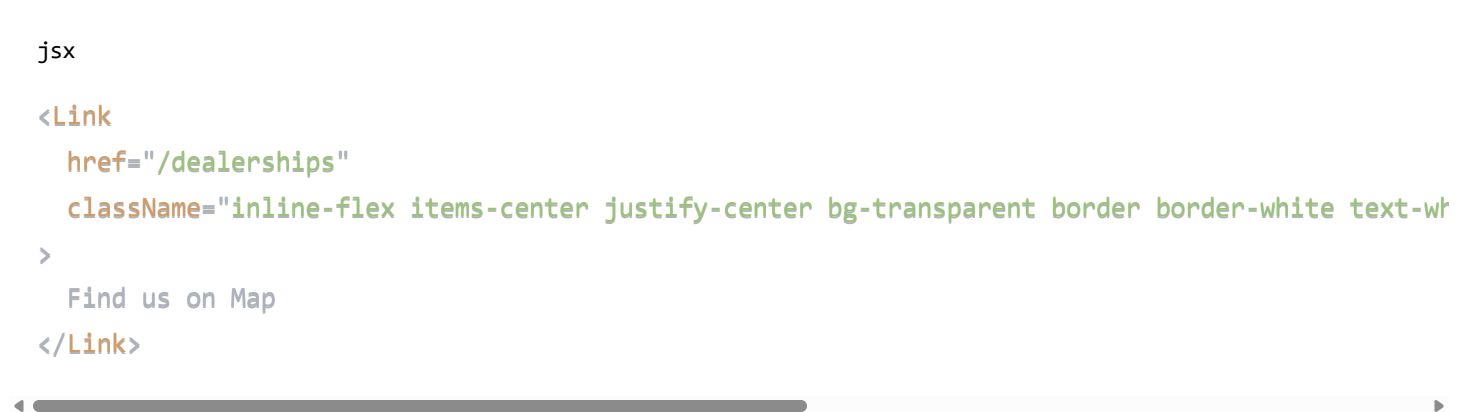
## Components

### Buttons

#### Primary Button



#### Secondary Button



#### Tertiary Button (Text Link)

jsx

```
<Link
  href="/news"
  className="group inline-flex items-center text-sm font-medium text-gray-900 tracking-wider hc
>
  VIEW ALL NEWS
  <svg
    xmlns="http://www.w3.org/2000/svg"
    className="h-4 w-4 ml-2 transform group-hover:translate-x-1 transition-transform duration-2
    fill="none"
    viewBox="0 0 24 24"
    stroke="currentColor"
  >
    <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={2} d="M14 5l7 7m0 0l7 7m-
  </svg>
</Link>
```

## Cards

### News Card

jsx

```
<article className="group flex flex-col h-full">
  {/* Image container with gradient overlay */}
  <div className="relative aspect-[16/9] overflow-hidden">
    <Image
      src={news.image}
      alt={news.title}
      fill
      sizes="(max-width: 768px) 100vw, 33vw"
      className="object-cover transition-transform duration-500 group-hover:scale-105"
    />

    {/* Gradient overlay that appears from bottom to top on hover */}
    <div className="absolute inset-0 bg-gradient-to-t from-primary/80 via-primary/40 to-transparent" />

    {/* Category Label */}
    <div className="absolute bottom-0 left-0 bg-white py-1 px-3 z-10">
      <span className="text-xs font-medium uppercase tracking-wider text-gray-900">{category}</span>
    </div>

    {/* New tag - flat design */}
    {isNew && (
      <div className="absolute top-0 right-0 bg-primary py-1 px-3 z-10">
        <span className="text-xs font-medium uppercase tracking-wider text-white">New</span>
      </div>
    )}
  </div>

  <div className="flex flex-col flex-grow p-5 border-1 border-r border-b border-gray-200 bg-white">
    {/* Content here */}
  </div>
</article>
```



**Info Card**

jsx

```
<div className="bg-white/10 backdrop-blur-sm border-1-2 border-primary p-4">
  <div className="flex items-center mb-2">
    <svg xmlns="http://www.w3.org/2000/svg" className="h-5 w-5 text-primary mr-2" fill="none" \
      <path strokeLinecap="round" strokeLinejoin="round" strokeWidth={1.5} d="M3 817.89 5.26a2
    </svg>
    <span className="text-white font-medium">Email Us</span>
  </div>
  <a href="mailto:info@cherybd.com" className="text-white/80 hover:text-white transition-colors
    info@cherybd.com
  </a>
</div>
```

## Navigation

### Footer Navigation

jsx

```
<div className="col-span-1 md:col-span-1 lg:col-span-2">
  <h4 className="text-gray-900 font-medium mb-5 pb-2 border-b border-gray-100">Category Title</
  <ul className="space-y-3">
    <li>
      <Link href="/path" className="text-gray-600 hover:text-primary transition-colors text-sm"
        Link Item
      </Link>
    </li>
    {/* More items */}
  </ul>
</div>
```

## Section Headers

jsx

```
<div>  
  <div className="w-10 h-1 bg-primary mb-4"></div>  
  <h2 className="text-2xl md:text-3xl font-bold text-gray-900 mb-2">  
    Section Title  
  </h2>  
  <p className="text-gray-600 max-w-lg">  
    Descriptive text about the section content  
  </p>  
</div>
```

## Animation Guidelines

### Transitions

- Use consistent durations:
  - Fast: `duration-200` (200ms)
  - Medium: `duration-300` (300ms)
  - Slow: `duration-500` (500ms)
- Common transition properties:
  - Colors: `transition-colors`
  - Transform: `transition-transform`
  - Opacity: `transition-opacity`
  - All: `transition-all` (use sparingly)

### Framer Motion Patterns

- Fade and rise in:



jsx

```
const contentVariants = {
  hidden: { opacity: 0, y: 30 },
  visible: {
    opacity: 1,
    y: 0,
    transition: {
      duration: 0.5,
      ease: "easeOut"
    }
  }
}

<motion.div
  variants={contentVariants}
  initial="hidden"
  animate={isInView ? "visible" : "hidden"}
>
  {/* Content */}
</motion.div>
```

- Staggered children:

jsx

```
const containerVariants = {
  hidden: { opacity: 0 },
  visible: {
    opacity: 1,
    transition: {
      staggerChildren: 0.15
    }
  }
};

const itemVariants = {
  hidden: { y: 30, opacity: 0 },
  visible: {
    y: 0,
    opacity: 1,
    transition: {
      type: "tween",
      duration: 0.5
    }
  }
};

<motion.div
  variants={containerVariants}
  initial="hidden"
  whileInView="visible"
  viewport={{ once: true, amount: 0.1 }}
>
  {items.map(item => (
    <motion.div key={item.id} variants={itemVariants}>
      {/* Item content */}
    </motion.div>
  ))}
</motion.div>
```

## Responsive Design

### Breakpoints

Follow Tailwind CSS default breakpoints:

- **Small (sm):** 640px and up

- **Medium (md):** 768px and up
- **Large (lg):** 1024px and up
- **Extra Large (xl):** 1280px and up
- **2xl:** 1536px and up

## Mobile-First Approach

- Start with mobile design and add complexity at larger breakpoints
- Stack elements vertically on mobile, arrange horizontally on larger screens
- Hide non-essential elements on mobile with `hidden md:block`
- Show mobile-specific elements with `block md:hidden`

## Common Responsive Patterns

jsx

*// Typography scaling*

```
<h2 className="text-2xl md:text-3xl lg:text-4xl">Heading</h2>
```

*// Grid adjustments*

```
<div className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4">
```

*// Padding/margin adjustments*

```
<section className="py-8 md:py-12 lg:py-16">
```

*// Flexbox direction changes*

```
<div className="flex flex-col md:flex-row">
```

## Image Guidelines

### Image Quality

- Use `quality={90}` for hero images and important product shots
- Use `quality={80}` for standard images
- Use `quality={70}` for background images or less critical visuals

### Aspect Ratios

- Hero banners: `aspect-[16/9]` or `aspect-[21/9]`
- News thumbnails: `aspect-[16/9]`

- Vehicle thumbnails: `aspect-[4/3]`
- Square elements: `aspect-square`

## Responsive Images

Always use proper image optimization:

```
jsx<Image  src="/image-path.jpg"  alt="Descriptive alt text"  fill  className="object-cover"  sizes="(max-width: 768px) 100vw, (max-width: 1200px) 50vw, 33vw"  priority={isHero} // Only for above-the-fold images />
```

## Accessibility Guidelines

### Color Contrast

- Ensure text has sufficient contrast against backgrounds (4.5:1 for normal text, 3:1 for large text)
- Don't rely solely on color to convey information
- Test with grayscale to verify sufficient contrast

### Interactive Elements

- Buttons and links should have visible focus states
- Minimum touch target size of 44x44px for mobile
- Use `aria-label` for icons without visible text

### Keyboard Navigation

- Ensure all interactive elements are keyboard accessible
- Maintain logical tab order
- Visible focus states that are consistent with design language

## Implementation Notes

### Tailwind Configuration

The design system assumes a Tailwind configuration with:

```
js
// tailwind.config.js
module.exports = {
  theme: {
    extend: {
      colors: {
        primary: '#D10000',
        'primary-dark': '#A60000',
        // Other custom colors...
      },
      // Other theme extensions...
    },
  },
  // Other config options...
}
```

## Component Structure

- Keep components focused on a single responsibility
- Use composition to build complex UIs from simple components
- Extract repeated patterns into reusable components

## Best Practices

### 1. Maintain Visual Hierarchy

- Use size, color, and spacing to guide users' attention
- Most important elements should be most prominent
- Create clear reading paths through content

### 2. Optimize Performance

- Lazy load below-the-fold images
- Use `priority` attribute only for critical above-the-fold images
- Use `whileInView` with `viewport={{ once: true }}` for animations

### 3. Mobile Considerations

- Ensure touch targets are at least 44px tall/wide
- Test on real devices when possible
- Consider network performance and optimize accordingly

### 4. Consistency

- Follow these guidelines for all new components
- Update existing components to match when feasible
- Maintain a "single source of truth" for design patterns