# FRONTEND GUIDELINES

## Bloom Academia

*Design System, Styling Standards & Performance Best Practices*

## INTRODUCTION

This document is the single source of truth for all frontend development on the AI-powered school platform. Following these guidelines ensures consistent styling, optimal performance, and a cohesive user experience across the entire application.

**Important:** Every developer must read and follow this guide. When in doubt, refer back to this document.

# 1. FRONTEND STACK

## 1.1 Core Technologies

| Technology | Version | Purpose |
|---|---|---|
| Next.js | 15.x | React framework with SSR/SSG |
| React | 18.x | UI component library |
| TypeScript | 5.x | Type-safe JavaScript |
| Tailwind CSS | 3.x | Utility-first CSS framework |

## 1.2 Required UI Libraries

Install these libraries at project setup. DO NOT skip any of these:

**Command to install all at once:**

```
pnpm install tailwindcss@latest framer-motion zustand react-konva konva clsx
tailwind-merge lucide-react @supabase/supabase-js
```

**Then install shadcn/ui:**

```
npx shadcn-ui@latest init
```

```
npx shadcn-ui@latest add button card progress dialog input label
```

| Library | Purpose |
|---|---|
| shadcn/ui | Pre-built accessible UI components (Button, Dialog, Card, etc.) |
| Framer Motion | Animation library for smooth transitions and effects |
| react-konva | Canvas library for interactive whiteboard rendering |
| Zustand | Lightweight state management |
| lucide-react | Icon library (clean, consistent icons) |
| clsx | Conditional className composition utility |
| tailwind-merge | Merge Tailwind classes intelligently (resolves conflicts) |

## 2. DESIGN SYSTEM

### 2.1 Color Palette

These are the ONLY colors allowed in the app. Do not use arbitrary hex codes.

#### Primary Colors

| Color Name | Hex Code | Tailwind Class | Usage |
|---|---|---|---|
| **Primary Blue** | #4A90E2 | bg-primary, text-primary | Primary buttons, links |
| **Success Green** | #7ED321 | bg-success, text-success | Success states, checkmarks |
| **Accent Orange** | #F5A623 | bg-accent, text-accent | Highlights, emphasis |
| **Error Red** | #E74C3C | bg-error, text-error | Error messages, warnings |

#### Neutral Colors

| Color Name | Hex Code | Tailwind Class | Usage |
|---|---|---|---|
| White | #FFFFFF | bg-white, text-white | Backgrounds, cards |
| Off-White | #FAFAFA | bg-gray-50 | Whiteboard background |
| Light Gray | #E5E7EB | bg-gray-200 | Borders, dividers |
| Medium Gray | #9CA3AF | bg-gray-400, text-gray-400 | Secondary text, icons |
| Dark Gray | #374151 | bg-gray-700, text-gray-700 | Primary text |
| Almost Black | #1F2937 | bg-gray-800, text-gray-800 | Headings, emphasis |

**Tailwind Config (tailwind.config.js):**

```
module.exports = {
  theme: {
    extend: {
      colors: {
        primary: '#4A90E2',
        success: '#7ED321',
        accent: '#F5A623',
        error: '#E74C3C',
      }
    }
  }
}
```

## 2.2 Typography

### Font Families

**Primary Font (Headings):** Inter

- Modern, clean, highly readable
- Use for: H1, H2, H3, buttons, nav
- Weights: 600 (semibold), 700 (bold)

**Secondary Font (Body):** System Font Stack

- -apple-system, BlinkMacSystemFont, 'Segoe UI', 'Roboto', 'Oxygen', 'Ubuntu'
- Fast loading (no font download)
- Use for: Body text, paragraphs, UI text
- Weights: 400 (normal), 500 (medium)

**Setup in layout.tsx:**

```
import { Inter } from 'next/font/google'


const inter = Inter({ subsets: ['latin'] })


export default function RootLayout({ children }) {
  return <html className={inter.className}>...</html>
}
```

### Text Sizes & Hierarchy

| Element | Size | Tailwind Class | Usage |
|---|---|---|---|
| H1 | 48px | text-5xl font-bold | Page titles, hero headlines |
| H2 | 36px | text-4xl font-semibold | Section headings |
| H3 | 24px | text-2xl font-semibold | Subsection headings |
| Body Large | 18px | text-lg | Important body text, intros |
| Body | 16px | text-base | Default body text, paragraphs |
| Body Small | 14px | text-sm | Secondary text, captions |
| Caption | 12px | text-xs | Fine print, labels |

## 2.3 Spacing & Layout

**8-Point Grid System:**
All spacing must be multiples of 8px (0.5rem in Tailwind).

| Size | Tailwind | Usage |
|---|---|---|
| 4px | p-1, m-1, gap-1 | Tight spacing (icon + text) |
| 8px | p-2, m-2, gap-2 | Small spacing (within components) |
| 16px | p-4, m-4, gap-4 | Default spacing (between elements) |
| 24px | p-6, m-6, gap-6 | Medium spacing (sections) |
| 32px | p-8, m-8, gap-8 | Large spacing (major sections) |
| 48px | p-12, m-12, gap-12 | Extra large spacing (page sections) |

**Border Radius:**
- Small: rounded-sm (2px) - inputs, tags
- Medium: rounded-md (6px) - buttons, cards
- Large: rounded-lg (8px) - modals, large cards
- Extra Large: rounded-xl (12px) - hero sections
- Full: rounded-full (9999px) - avatars, pills

**Shadows:**
- Small: shadow-sm - subtle elevation
- Medium: shadow-md - cards, buttons
- Large: shadow-lg - modals, dropdowns
- Extra Large: shadow-2xl - hero elements

# 3. COMPONENT GUIDELINES

## 3.1 Buttons

### Primary Button:

```
<Button className="bg-primary hover:bg-primary/90 text-white font-semibold">
  Start Learning
</Button>
```

### Secondary Button:

```
<Button variant="outline" className="border-gray-300 hover:bg-gray-50">
  Learn More
</Button>
```

### Success Button:

```
<Button className="bg-success hover:bg-success/90 text-white">
  Complete Lesson
</Button>
```

### Button Sizes:

- Small: className="h-9 px-3 text-sm"
- Medium (default): className="h-10 px-4"
- Large: className="h-12 px-6 text-lg"

## 3.2 Cards

```
<Card className="p-6 border-gray-200 shadow-md hover:shadow-lg transition-
shadow">
  <CardHeader>
    <CardTitle className="text-2xl font-semibold">Lesson Title</CardTitle>
  </CardHeader>
  <CardContent>
    <p className="text-gray-600">Lesson description...</p>
  </CardContent>
</Card>
```

## 3.3 Loading States

### Skeleton Loader (use for cards, text):

```
<div className="animate-pulse">
```

```
  <div className="h-4 bg-gray-200 rounded w-3/4 mb-2"></div>
  <div className="h-4 bg-gray-200 rounded w-1/2"></div>
</div>
```

### Spinner (use for buttons):

```
import { Loader2 } from 'lucide-react'

<Button disabled>
  <Loader2 className="mr-2 h-4 w-4 animate-spin" />
  Loading...
</Button>
```

# 4. PERFORMANCE OPTIMIZATION

## 4.1 Image Optimization

### ALWAYS use Next.js Image component:

```
import Image from 'next/image'


<Image
  src="/hero-image.jpg"
  alt="Student learning"
  width={800}
  height={600}
  priority // for above-the-fold images
  placeholder="blur" // optional: for smooth loading
/>
```

**Benefits:**

- Automatic lazy loading
- Image format optimization (WebP, AVIF)
- Responsive images
- No layout shift

## 4.2 Code Splitting & Lazy Loading

### Lazy load heavy components:

```
import dynamic from 'next/dynamic'


const Whiteboard = dynamic(() => import('@/components/Whiteboard'), {
  loading: () => <div>Loading whiteboard...</div>,
  ssr: false // if component uses browser APIs
})
```

**When to use:**

- Heavy visualizations (whiteboard canvas)
- Components not needed on initial load
- Modal/dialog content
- Components that use browser-only APIs

## 4.3 Font Optimization

**Use Next.js font optimization (already configured):**

- Fonts are self-hosted (no external requests)
- Automatic font subsetting
- Preloaded on page load

**CRITICAL: Add font-display swap:**

```
const inter = Inter({
  subsets: ['latin'],
  display: 'swap' // prevents invisible text during load
})
```

## 4.4 Bundle Size Optimization

**Import only what you need:**

```
// ❌ BAD - imports entire library
import * as Icons from 'lucide-react'


// ✅ GOOD - tree-shakeable
import { Mic, Play, Pause } from 'lucide-react'
```

**Check bundle size:**

```
pnpm build
# Look for pages that are >200KB
```

## 4.5 Loading Speed Targets

| Metric | Target | How to Measure |
|---|---|---|
| First Contentful Paint (FCP) | < 1.8 seconds | Lighthouse / Chrome DevTools |
| Largest Contentful Paint (LCP) | < 2.5 seconds | Lighthouse / Chrome DevTools |
| Time to Interactive (TTI) | < 3.8 seconds | Lighthouse |
| Total Bundle Size | < 200KB (gzipped) | Next.js build output |

## 5. RESPONSIVE DESIGN

### 5.1 Breakpoints

| Device | Breakpoint | Tailwind Prefix |
|---|---|---|
| Mobile | < 640px | (default, no prefix) |
| Tablet | ≥ 640px | sm: |
| Laptop | ≥ 1024px | lg: |
| Desktop | ≥ 1280px | xl: |

**Mobile-First Approach:**
Always design for mobile first, then scale up.

**Example:**

```
<div className="p-4 sm:p-6 lg:p-8">
  <h1 className="text-2xl sm:text-3xl lg:text-5xl">
    Welcome
  </h1>
</div>
```

## 6. ACCESSIBILITY (A11Y)

### 6.1 Color Contrast

**All text must meet WCAG AA standards:**

- Normal text (< 18px): Contrast ratio ≥ 4.5:1
- Large text (≥ 18px): Contrast ratio ≥ 3:1
- Use WebAIM Contrast Checker to verify

### 6.2 Focus States

**All interactive elements MUST have visible focus:**

```
<button className="focus:outline-none focus:ring-2 focus:ring-primary focus:ring-offset-2">
  Click Me
</button>
```

### 6.3 ARIA Labels

**Add labels for screen readers:**

```
<button aria-label="Close dialog">
  <X className="h-4 w-4" />
</button>

<img src="/logo.png" alt="AI School Logo" />
```

### 6.4 Keyboard Navigation

**Ensure all functionality is keyboard-accessible:**

- Tab: Navigate forward
- Shift+Tab: Navigate backward
- Enter/Space: Activate buttons/links
- Escape: Close modals/dialogs

# 7. CODING BEST PRACTICES

## 7.1 Component Structure

```
// components/Button.tsx
import { ButtonHTMLAttributes, forwardRef } from 'react'
import { cn } from '@/lib/utils'

interface ButtonProps extends ButtonHTMLAttributes<HTMLButtonElement> {
  variant?: 'primary' | 'secondary' | 'outline'
  size?: 'sm' | 'md' | 'lg'
}

const Button = forwardRef<HTMLButtonElement, ButtonProps>(
  ({ className, variant = 'primary', size = 'md', ...props }, ref) => {
    return (
      <button
        ref={ref}
        className={cn(
          'rounded-md font-semibold transition-colors',
          variant === 'primary' && 'bg-primary text-white hover:bg-
primary/90',
          size === 'md' && 'h-10 px-4',
          className
        )}
        {...props}
      />
    )
  }
)
```

## 7.2 File Organization

```
app/
├── (routes)/
│   ├── page.tsx              # Landing page
│   ├── learn/page.tsx        # Learning interface
│   └── lessons/page.tsx      # Lesson selection
```

```
├── api/                    # API routes
├── layout.tsx              # Root layout
└── globals.css             # Global styles

components/
├── ui/                     # shadcn components
├── Whiteboard.tsx
├── VoiceIndicator.tsx
└── LessonCard.tsx

lib/
├── utils.ts                # Utility functions
├── supabase.ts             # Supabase client
└── store.ts                # Zustand store
```

### 7.3 Naming Conventions

- Components: PascalCase (Button.tsx, LessonCard.tsx)
- Files/folders: kebab-case (lesson-selection/, api-utils.ts)
- Functions: camelCase (handleClick, fetchLessons)
- Constants: UPPER_SNAKE_CASE (API_URL, MAX_RETRIES)
- CSS classes: Use Tailwind only (no custom classes in most cases)

## 8. QUICK REFERENCE CHECKLIST

**Before deploying, ensure:**

✅ Design System
- Using only approved colors from palette
- Typography follows size hierarchy
- Spacing uses 8-point grid
- Consistent border radius and shadows

✅ Performance
- All images use Next.js Image component
- Heavy components are lazy loaded
- Bundle size < 200KB
- Lighthouse score > 90

✅ Responsive
- Tested on mobile, tablet, desktop
- No horizontal scrolling
- Touch targets ≥ 44x44px on mobile

✅ Accessibility
- Color contrast meets WCAG AA
- All interactive elements have focus states
- ARIA labels on icons/buttons
- Keyboard navigation works

✅ Code Quality
- No console.log in production
- TypeScript errors resolved
- ESLint passes
- Components properly typed

**Follow these guidelines religiously!**

** * * END OF FRONTEND GUIDELINES * * **