

RailLynk Design Manual

Comprehensive UI/UX Design System & Guidelines

Version: 2.0

Last Updated: July 22, 2025

Project: RailLynk Railway Management System



Table of Contents

- [1. Design Overview](#)
- [2. Brand Identity](#)
- [3. Color System](#)
- [4. Typography](#)
- [5. Layout System](#)
- [6. Component Design Patterns](#)
- [7. Icon System](#)
- [8. UI Component Library](#)
- [9. Responsive Design](#)
- [10. Animation & Interactions](#)
- [11. Accessibility Guidelines](#)
- [12. Design Tokens](#)
- [13. User Interface Patterns](#)
- [14. Implementation Guidelines](#)
- [15. Quality Assurance](#)
- [16. Testing Manual](#)



Design Overview

Design Philosophy

RailLynk follows a **modern, clean, and professional design approach** that prioritizes:

- **User Experience:** Intuitive navigation and clear information hierarchy
- **Accessibility:** WCAG 2.1 compliant design for all users
- **Consistency:** Unified visual language across all interfaces
- **Performance:** Optimized for fast loading and smooth interactions
- **Scalability:** Design system that grows with the platform

Design Principles

1. **Clarity:** Clear visual hierarchy and information organization
2. **Efficiency:** Streamlined workflows and minimal friction
3. **Trust:** Professional appearance that instills confidence
4. **Accessibility:** Universal design for all users
5. **Consistency:** Unified experience across all touchpoints

Target Users

- **Railway Passengers:** Consumer-friendly interface for ticketing and tracking
 - **Station Masters:** Professional dashboard for operational management
 - **System Administrators:** Comprehensive control panels with data visualization
-



Brand Identity

Brand Colors

Primary Brand Colors:

- **Railway Blue:** #1e3a8a - Primary brand color, represents trust and reliability
- **Railway Green:** #059669 - Secondary color, represents sustainability and progress
- **Railway Red:** #dc2626 - Accent color for alerts and critical actions

Extended Brand Palette:

- **Deep Navy:** #1e2a3a - For headers and premium sections
- **Sky Blue:** #3b82f6 - Interactive elements and links
- **Light Blue:** #e5f3ff - Background accents and hover states

Logo & Branding

- **Primary Logo:** Train icon with "RailLynk" wordmark
- **Icon Only:** Simplified train symbol for small spaces
- **Color Variations:** White, dark, and monochrome versions available

Voice & Tone

- **Professional:** Clear, authoritative communication
 - **Helpful:** Supportive and informative messaging
 - **Modern:** Contemporary language without jargon
 - **Trustworthy:** Reliable and consistent information
-

Primary Colors (Actual Implementation)

```
:root {  
  /* Primary Brand Colors - From index.css */  
  --bar-color: rgb(16, 43, 107);          /* Primary brand blue */  
  --primary-text-color: rgb(35, 66, 116); /* Primary text blue */  
  --secondary-text-color: rgb(255, 255, 255); /* White text */  
  --primary-color: #0056b3;              /* Bootstrap-style primary */  
  
  /* Semantic Colors */  
  --success-color: #28a745;              /* Green success */  
  --danger-color: #dc3545;               /* Red danger */  
  --warning-color: #ffc107;              /* Yellow warning */  
  --info-color: #17a2b8;                 /* Cyan info */  
  --light-color: #f8f9fa;                /* Light gray */  
  --dark-color: #343a40;                 /* Dark gray */  
  
  /* Layout Colors */  
  --sidebar-bg: #1e2a3a;                 /* Dark sidebar background */  
  --sidebar-text: #ffffff;                /* White sidebar text */  
}
```

Background Colors

```
/* Main application background */  
body {  
  background-color: #f5f7fb; /* Light blue-gray background */  
  color: #333;  
}  
  
/* Card backgrounds */  
.announcement-card {  
  background-color: #c1f5fe; /* Light cyan for announcement cards */  
  box-shadow: #c1f5fe 0px 0px 10px;  
}
```

Gradient Systems

From Login Component:

```

/* Primary gradient - Blue theme */
.login-card {
  background: linear-gradient(135deg, #667eea 0%, #4b5ba2 100%);
  box-shadow:
    0 25px 50px rgba(102, 126, 234, 0.5),
    0 15px 30px rgba(0, 0, 0, 0.2);
}

/* Dashboard balance card gradient */
.balance-icon {
  background: linear-gradient(45deg, #1e3a8a, #3b82f6);
}

```

Color Usage Guidelines

Primary Blue (`rgb(16, 43, 107)`)

- **Usage:** Navigation bars, sidebars, primary buttons
- **Implementation:** `--bar-color` variable used throughout
- **Components:** Navbar, Sidebar, Admin Dashboard

Secondary Blue (`#0056b3`)

- **Usage:** Interactive elements, links, secondary actions
- **Context:** Form buttons, card highlights
- **Accessibility:** High contrast with white text

Announcement Card Theme (`#c1f5fe`)

- **Usage:** Information cards, notification components
- **Context:** Announcement cards with light cyan background
- **Visual Impact:** Soft, approachable information display



Typography

Font Stack (Actual Implementation)

```

/* Primary Font Family - From index.css */
:root {
  --font-family: "Poppins", sans-serif;
}

/* Google Fonts Import */
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&display=swap');

/* Fallback Font Stack */
body {

```

```

font-family: 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', sans-serif;
line-height: 1.6;
}

/* Modern Font Stack for Dashboards */
.dashboard-container {
  font-family: 'Inter', -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif;
}

```

Typography Hierarchy

```

/* Dashboard Titles */
.dashboard-title {
  font-size: 2rem;           /* 32px - Main dashboard title */
  font-weight: 600;
  color: #2c3e50;
}

.dashboard-title (passenger) {
  font-size: 2.5rem;         /* 40px - Passenger dashboard */
  font-weight: 700;
  background: linear-gradient(45deg, #ffffff, #e5f3ff);
  -webkit-background-clip: text;
  background-clip: text;
}

/* Card and Component Typography */
.login-card h2 {
  font-size: 24px;           /* Login form title */
  font-weight: 700;
  letter-spacing: 2px;
  color: white;
}

/* Sidebar Menu Text */
.menu-item {
  font-size: 1rem;           /* 16px - Navigation items */
  font-weight: normal;
}

```

Font Weights (Available in Poppins)

```

/* Poppins Weight Variants */
--font-weight-light: 300;    /* Light - Subtle text */
--font-weight-regular: 400;  /* Regular - Body text */
--font-weight-semibold: 600; /* Semibold - Headings, emphasis */

/* Usage Examples */
.dashboard-subtitle {
  font-weight: 400;           /* Regular for subtitles */
}

```

```
.dashboard-title {
  font-weight: 600;           /* Semibold for titles */
}

.login-card h2 {
  font-weight: 700;           /* Bold for login headers */
}
```

Responsive Typography

```
/* Mobile-first Typography */
@media (max-width: 768px) {
  .dashboard-title {
    font-size: 1.5rem;        /* Smaller on mobile */
  }

  .login-card h2 {
    font-size: 20px;          /* Reduced login title */
  }
}

/* Large Screen Enhancements */
@media (min-width: 1200px) {
  .dashboard-title {
    font-size: 2.5rem;        /* Larger on desktop */
  }
}
```

Text Color Implementation

```
/* Actual color usage from stylesheets */
h2 {
  color: white;               /* Default heading color */
}

.dashboard-title {
  color: #2c3e50;             /* Dark blue-gray for admin */
}

.dashboard-subtitle {
  color: #e5f3ff;             /* Light blue for passenger dashboard */
  opacity: 0.9;
}

/* Announcement Card Text */
.announcement-card > h2 {
  color: #000;                /* Black for card headings */
}

.announcement-card > span {
```

```
color: #434343;           /* Gray for date stamps */
font-size: 14px;
}
```

Layout System

Grid System (Actual Implementation)

RailLynk uses **CSS Flexbox** and **CSS Grid** for responsive layouts:

```
/* Base Container System - From index.css */
:root {
  --sidebar-width: 220px;
  --sidebar-collapsed-width: 60px;
  --header-height: 60px;
  --transition-speed: 0.3s;
}

/* Dashboard Layout Pattern */
.dashboard-container {
  display: flex;
  min-height: 100vh;
  background-color: #f5f7fb;
  font-family: 'Segoe UI', Roboto, Oxygen, Ubuntu, Cantarell, 'Open Sans', sans-serif;
}
```

Sidebar Layout System

```
/* Fixed Sidebar Implementation */
.sidebar {
  background-color: var(--bar-color);
  position: fixed;
  top: 0;
  left: 0;
  height: 100vh;
  width: 60px;           /* Collapsed state */
  transition: width 0.3s ease;
  z-index: 100;
  overflow-x: hidden;
  padding-top: 10px;
}

.sidebar.open {
  width: 220px;          /* Expanded state */
  z-index: 1000;
}

/* Main Content Adjustment */
.main-content {
  flex: 1;
}
```

```
padding: 20px;
transition: margin-left 0.3s ease;
margin-left: 60px;          /* Space for collapsed sidebar */
}

.sidebar.open + .main-content {
margin-left: 220px;          /* Space for expanded sidebar */
}
```

Responsive Grid Patterns

Stats Cards Layout

```
/* Admin Dashboard Stats Row */
.stats-row {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(280px, 1fr));
gap: 20px;
margin-bottom: 30px;
}

.stat-card {
background: white;
padding: 20px;
border-radius: 12px;
box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05);
border: 1px solid #e5e7eb;
}
```

Form Layout Patterns

```
/* Login Form Layout */
.main-login-container {
min-height: 100vh;
display: flex;
justify-content: center;
align-items: center;
background-size: cover;
background-position: center;
padding: 20px;
}

.login-card {
width: 100%;
max-width: 600px;
padding: 24px 32px;
border-radius: 24px;
text-align: center;
}
```


Spacing System (From CSS Variables)

```
:root {
  --border-radius: 8px;           /* Standard border radius */
  --box-shadow: 0 4px 12px rgba(0, 0, 0, 0.05); /* Standard shadow */
}

/* Consistent spacing patterns */
.dashboard-header {
  margin-bottom: 30px;           /* Section spacing */
}

.menu-item {
  padding: 15px;                 /* Menu item padding */
}

.main-content {
  padding: 20px;                 /* Content padding */
}

/* Passenger Dashboard spacing */
.main-content (passenger) {
  padding: 2rem;                 /* Larger padding for passenger views */
}
```

Navigation Layout

Navbar Structure

```
.navbar {
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 15px 20px;
  background: var(--bar-color);
  position: relative;
  z-index: 2000;
}

/* Mobile Navigation */
@media (max-width: 768px) {
  .nav-links {
    display: none;
    position: absolute;
    top: 60px;
    right: 0;
    background: var(--bar-color);
    width: 150px;
    flex-direction: column;
  }

  .nav-links.open {
```

```
    display: flex;
  }
}
```

Card Layout System

Announcement Cards

```
.announcement-card {
  display: flex;
  flex-direction: column;
  justify-content: space-between;
  padding: 10px;
  width: 250px;
  border-radius: 0.5rem;
  box-shadow: #c1f5fe 0px 0px 10px;
}
```

Responsive Breakpoints

```
/* Mobile-first responsive design */
@media (max-width: 768px) {
  .sidebar.open {
    width: 220px;
  }

  .main-content {
    margin-left: 60px;
  }

  .dashboard-title {
    font-size: 1.5rem;          /* Smaller titles on mobile */
  }

  .hamburger {
    display: block;            /* Show mobile menu */
  }
}

/* Large screens */
@media (min-width: 1200px) {
  .dashboard-title {
    font-size: 2.5rem;          /* Larger titles on desktop */
  }
}
```

Layout Components Hierarchy

Dashboard Structure

```
dashboard-container (flex)
├─ sidebar (fixed, collapsible)
│   └─ menu-button
│   └─ menu-list
│       └─ menu-item (multiple)
└─ main-content (flex: 1)
    └─ dashboard-header
    └─ stats-row (grid)
        └─ stat-card (multiple)
    └─ content-sections
```

Form Structure

```
main-login-container (flex, full height)
└─ login-card (centered, max-width)
    └─ login-icon
    └─ title (h2)
    └─ input-group (multiple)
```

Z-Index Management

```
/* Z-index hierarchy */
.navbar {
  z-index: 2000;           /* Highest - Always visible */
}

.sidebar.open {
  z-index: 1000;          /* High - Above content */
}

.sidebar {
  z-index: 100;           /* Medium - Base sidebar */
}

.login-card {
  z-index: 10;            /* Above background overlay */
}
```

Component Design Patterns

Card Component Pattern

```
.card {
  background: var(--surface-elevated);
  border-radius: var(--border-radius-lg);
  box-shadow: var(--shadow-sm);
  border: 1px solid var(--border-color);
  overflow: hidden;
  transition: all var(--transition-duration) ease;
}

.card:hover {
  box-shadow: var(--shadow-md);
  transform: translateY(-2px);
}

.card-header {
  background: linear-gradient(45deg, var(--primary-blue), var(--sky-blue));
  color: var(--text-inverse);
  padding: var(--space-6) var(--space-8);
}

.card-content {
  padding: var(--space-8);
}

.card-footer {
  padding: var(--space-6) var(--space-8);
  background: var(--background-secondary);
  border-top: 1px solid var(--border-color);
}
```

Button Component Patterns

```
/* Base Button */
.btn {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  gap: var(--space-2);
  padding: var(--space-3) var(--space-6);
  border: 1px solid transparent;
  border-radius: var(--border-radius);
  font-family: inherit;
  font-size: var(--font-size-base);
  font-weight: var(--font-weight-medium);
  line-height: 1;
  cursor: pointer;
  transition: all var(--transition-duration) ease;
```

```

text-decoration: none;
white-space: nowrap;
}

/* Primary Button */
.btn-primary {
  background: linear-gradient(45deg, var(--primary-blue), var(--sky-blue));
  color: var(--text-inverse);
  box-shadow: var(--shadow-sm);
}

.btn-primary:hover {
  background: linear-gradient(45deg, var(--royal-blue), var(--primary-blue));
  box-shadow: var(--shadow-md);
  transform: translateY(-1px);
}

/* Secondary Button */
.btn-secondary {
  background: var(--background-primary);
  color: var(--primary-blue);
  border-color: var(--primary-blue);
}

.btn-secondary:hover {
  background: var(--primary-blue);
  color: var(--text-inverse);
}

/* Size Variants */
.btn-sm { padding: var(--space-2) var(--space-4); font-size: var(--font-size-sm); }
.btn-lg { padding: var(--space-4) var(--space-8); font-size: var(--font-size-lg); }
.btn-xl { padding: var(--space-5) var(--space-10); font-size: var(--font-size-xl); }

```

Form Component Patterns

```

/* Form Groups */
.form-group {
  margin-bottom: var(--space-6);
}

.form-label {
  display: flex;
  align-items: center;
  gap: var(--space-2);
  font-size: var(--font-size-sm);
  font-weight: var(--font-weight-medium);
  color: var(--text-secondary);
  margin-bottom: var(--space-2);
  text-transform: uppercase;
  letter-spacing: 0.05em;
}

```

```

.form-input {
  width: 100%;
  padding: var(--space-4) var(--space-4);
  border: 1px solid var(--border-color);
  border-radius: var(--border-radius);
  font-family: inherit;
  font-size: var(--font-size-base);
  background: var(--background-primary);
  transition: all var(--transition-duration) ease;
}

.form-input:focus {
  outline: none;
  border-color: var(--primary-blue);
  box-shadow: 0 0 0 3px rgba(30, 64, 175, 0.1);
  background: var(--background-primary);
}

/* Error States */
.form-input.error {
  border-color: var(--error-color);
  background: rgba(239, 68, 68, 0.05);
}

.form-error {
  color: var(--error-color);
  font-size: var(--font-size-sm);
  margin-top: var(--space-1);
}

```

Icon System

Icon Library (Actual Implementation)

RailLynk uses **React Icons** (v5.5.0) primarily from the Feather Icons collection:

```

// Primary import pattern used throughout the application
import {
  FiUsers, FiSettings, FiMenu, FiUserPlus, FiMapPin,
  FiList, FiMonitor, FiLogOut, FiBarChart2,
  FiTrendingUp, FiCalendar, FiDollarSign,
  FiCreditCard, FiUser, FiDatabase, FiPause,
  FiActivity
} from "react-icons/fi";

// FontAwesome icons for specific use cases
import { FaTicketAlt } from "react-icons/fa";

```

Core Icon Set (From Actual Components)

Navigation Icons

```
// Sidebar and menu icons
<FiMenu size={24} />           // Hamburger menu
<FiUser size={20} />           // Profile/user icon
<FiSettings size={20} />       // Settings gear
<FiLogout size={20} />         // Logout arrow
<FiMapPin size={20} />         // Location/station marker
```

Dashboard Icons

```
// Admin dashboard metrics
<FiUsers />                     // Total passengers
<FiCreditCard />               // Cards issued
<FiDollarSign />               // Revenue
<FiTrendingUp />               // Growth indicators
<FiBarChart2 />                // Analytics
<FiCalendar />                 // Date/time
<FiActivity />                 // Real-time data
```

Feature-Specific Icons

```
// Passenger dashboard
<FaTicketAlt />                // Smart tickets
<FiDatabase />                 // Data/transactions
<FiPause />                    // Status indicators

// Admin functions
<FiUserPlus />                 // Add user
<FiMonitor />                  // System monitoring
<FiList />                     // Lists/tables
```

Icon Usage Patterns

Sidebar Menu Implementation

```
// From PassengerDashboard.jsx
<button className="menu-item" onClick={() => navigate("/passengerProfile")}>
  <FiUser size={20} /> {isSidebarOpen && "Profile"}
</button>

<button className="menu-item" onClick={() => navigate("/passengerRecharge")}>
  <FiSettings size={20} /> {isSidebarOpen && "Settings"}
</button>

<button className="menu-item logout-button" onClick={handleLogout}>
  <FiLogout size={20} /> {isSidebarOpen && "Logout"}
</button>
```

Dashboard Statistics Icons

```
// From AdminDashboard.jsx - Stats cards with icons
const statsCards = [
  {
    icon: <FiCreditCard size={28} />,
    title: "Cards Issued Today",
    value: dashboardStats.totalCardsIssuedToday,
    color: "#4f46e5"
  },
  {
    icon: <FiUsers size={28} />,
    title: "Total Passengers",
    value: dashboardStats.totalPassengers,
    color: "#059669"
  }
];
```

Icon Sizing Standards

```
/* Size variations used in components */
size={20}    /* Standard sidebar menu items */
size={24}    /* Hamburger menu button */
size={28}    /* Dashboard stat cards */

/* Custom sizing in CSS */
.menu-item svg {
  color: white;
  margin-right: 15px;
}

.logout-button svg {
  color: #ff6b6b; /* Red color for logout */
}
```

Interactive Icon States

```
/* Hover states for icon buttons */
.menu-item:hover {
  background-color: rgba(255, 255, 255, 0.1);
  color: white;
}

.logout-button:hover {
  background-color: rgba(255, 0, 0, 0.1);
  color: #ff6b6b;
}

.menu-button:hover {
  background-color: rgba(255, 255, 255, 0.1);
}
```



```
}
```

Icon Color System

```
/* Default icon colors */
.sidebar .menu-item svg {
  color: white; /* White for sidebar icons */
}

.logout-button svg {
  color: #ff6b6b; /* Red for logout icon */
}

/* Dashboard stat card icons - Dynamic colors */
.stat-icon {
  color: #4f46e5; /* Blue for cards */
  color: #059669; /* Green for passengers */
  color: #dc2626; /* Red for alerts */
}
```

Conditional Icon Rendering

```
// Icons shown/hidden based on sidebar state
{isSidebarOpen && "Profile"} // Text shown when sidebar open
<FiUser size={20} /> // Icon always visible

// Hamburger menu for mobile
<div className="hamburger" onClick={toggleMenu}>
  ≡ { /* Simple Unicode character for mobile menu */ }
</div>
```

Accessibility Implementation

```
// Icons with proper ARIA labels
<button
  className="menu-button"
  onClick={toggleSidebar}
  aria-label="Toggle navigation menu"
>
  <FiMenu size={24} />
</button>

// Icons with descriptive text
<FiUser size={20} /> {isSidebarOpen && "Profile"}
```

Navigation Components

Primary Navigation

```
.navbar {
  background: linear-gradient(135deg, var(--deep-navy), var(--primary-blue));
  color: var(--text-inverse);
  padding: var(--space-4) 0;
  box-shadow: var(--shadow-lg);
  position: sticky;
  top: 0;
  z-index: 1000;
}

.navbar-brand {
  font-size: var(--font-size-xl);
  font-weight: var(--font-weight-bold);
  text-decoration: none;
  color: var(--text-inverse);
  display: flex;
  align-items: center;
  gap: var(--space-3);
}

.navbar-nav {
  display: flex;
  list-style: none;
  gap: var(--space-6);
  align-items: center;
}

.nav-link {
  color: rgba(255, 255, 255, 0.9);
  text-decoration: none;
  font-weight: var(--font-weight-medium);
  padding: var(--space-2) var(--space-4);
  border-radius: var(--border-radius);
  transition: all var(--transition-duration) ease;
}

.nav-link:hover,
.nav-link.active {
  color: var(--text-inverse);
  background: rgba(255, 255, 255, 0.1);
}
```

Sidebar Navigation

```
.sidebar {
  background: var(--deep-navy);
  width: 220px;
  height: 100vh;
  position: fixed;
  left: 0;
  top: 0;
  z-index: 1000;
  overflow-y: auto;
  transition: all var(--transition-duration) ease;
}

.sidebar.collapsed {
  width: 60px;
}

.sidebar-header {
  padding: var(--space-6);
  border-bottom: 1px solid rgba(255, 255, 255, 0.1);
}

.sidebar-nav {
  list-style: none;
  padding: var(--space-4) 0;
}

.sidebar-nav-item {
  margin-bottom: var(--space-1);
}

.sidebar-nav-link {
  display: flex;
  align-items: center;
  gap: var(--space-3);
  padding: var(--space-4) var(--space-6);
  color: rgba(255, 255, 255, 0.8);
  text-decoration: none;
  font-weight: var(--font-weight-medium);
  transition: all var(--transition-duration) ease;
}

.sidebar-nav-link:hover,
.sidebar-nav-link.active {
  background: rgba(255, 255, 255, 0.1);
  color: var(--text-inverse);
  border-right: 3px solid var(--sky-blue);
}
```

Data Display Components

Statistics Cards

```
.stats-grid {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
  gap: var(--space-6);
  margin-bottom: var(--space-8);
}

.stat-card {
  background: var(--surface-elevated);
  border-radius: var(--border-radius-lg);
  padding: var(--space-8);
  box-shadow: var(--shadow-sm);
  border: 1px solid var(--border-color);
  display: flex;
  align-items: center;
  gap: var(--space-4);
  transition: all var(--transition-duration) ease;
}

.stat-card:hover {
  transform: translateY(-4px);
  box-shadow: var(--shadow-md);
}

.stat-icon {
  background: linear-gradient(45deg, var(--primary-blue), var(--sky-blue));
  color: var(--text-inverse);
  padding: var(--space-4);
  border-radius: var(--border-radius-lg);
  display: flex;
  align-items: center;
  justify-content: center;
}

.stat-content h3 {
  font-size: var(--font-size-sm);
  font-weight: var(--font-weight-medium);
  color: var(--text-tertiary);
  margin: 0 0 var(--space-2) 0;
  text-transform: uppercase;
  letter-spacing: 0.05em;
}

.stat-value {
  font-size: var(--font-size-h2);
  font-weight: var(--font-weight-bold);
  color: var(--text-primary);
  margin: 0;
```

```

    line-height: 1;
  }

  .stat-change {
    font-size: var(--font-size-sm);
    font-weight: var(--font-weight-medium);
    margin-top: var(--space-1);
  }

  .stat-change.positive { color: var(--success-color); }
  .stat-change.negative { color: var(--error-color); }

```

Tables

```

.table-container {
  background: var(--surface-elevated);
  border-radius: var(--border-radius-lg);
  overflow: hidden;
  box-shadow: var(--shadow-sm);
  border: 1px solid var(--border-color);
}

.table {
  width: 100%;
  border-collapse: collapse;
}

.table thead {
  background: var(--background-secondary);
}

.table th {
  padding: var(--space-4) var(--space-6);
  text-align: left;
  font-weight: var(--font-weight-semibold);
  color: var(--text-secondary);
  font-size: var(--font-size-sm);
  text-transform: uppercase;
  letter-spacing: 0.05em;
  border-bottom: 1px solid var(--border-color);
}

.table td {
  padding: var(--space-4) var(--space-6);
  color: var(--text-primary);
  border-bottom: 1px solid var(--border-color);
}

.table tbody tr:hover {
  background: var(--background-secondary);
}

```

```
.table tbody tr:last-child td {  
  border-bottom: none;  
}
```

Interactive Components

Modal/Dialog

```
.modal-overlay {  
  position: fixed;  
  top: 0;  
  left: 0;  
  right: 0;  
  bottom: 0;  
  background: rgba(0, 0, 0, 0.5);  
  backdrop-filter: blur(4px);  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  z-index: 2000;  
  animation: fadeIn var(--transition-duration) ease;  
}  
  
.modal {  
  background: var(--surface-elevated);  
  border-radius: var(--border-radius-xl);  
  box-shadow: var(--shadow-xl);  
  max-width: 500px;  
  width: 90vw;  
  max-height: 90vh;  
  overflow: hidden;  
  animation: slideUp var(--transition-duration) ease;  
}  
  
.modal-header {  
  padding: var(--space-6) var(--space-8);  
  border-bottom: 1px solid var(--border-color);  
  display: flex;  
  align-items: center;  
  justify-content: space-between;  
}  
  
.modal-title {  
  font-size: var(--font-size-lg);  
  font-weight: var(--font-weight-semibold);  
  color: var(--text-primary);  
  margin: 0;  
}  
  
.modal-close {  
  background: none;  
  border: none;
```

```
padding: var(--space-2);
border-radius: var(--border-radius);
color: var(--text-tertiary);
cursor: pointer;
transition: all var(--transition-duration) ease;
}

.modal-close:hover {
  background: var(--background-secondary);
  color: var(--text-primary);
}

.modal-body {
  padding: var(--space-8);
}

.modal-footer {
  padding: var(--space-6) var(--space-8);
  border-top: 1px solid var(--border-color);
  display: flex;
  gap: var(--space-4);
  justify-content: flex-end;
}

@keyframes fadeIn {
  from { opacity: 0; }
  to { opacity: 1; }
}

@keyframes slideUp {
  from {
    opacity: 0;
    transform: translateY(20px) scale(0.95);
  }
  to {
    opacity: 1;
    transform: translateY(0) scale(1);
  }
}
```



Responsive Design

Breakpoints

```
:root {
  --breakpoint-xs: 480px;    /* Extra small devices */
  --breakpoint-sm: 640px;    /* Small devices */
  --breakpoint-md: 768px;    /* Medium devices */
  --breakpoint-lg: 1024px;   /* Large devices */
  --breakpoint-xl: 1280px;   /* Extra large devices */
  --breakpoint-2xl: 1536px;  /* 2X large devices */
}
```

```
}
```

Mobile-First Approach

```
/* Base (mobile) styles */
.container {
  padding: var(--space-4);
}

.card-grid {
  grid-template-columns: 1fr;
  gap: var(--space-4);
}

/* Tablet styles */
@media (min-width: 768px) {
  .container {
    padding: var(--space-8);
  }

  .card-grid {
    grid-template-columns: repeat(2, 1fr);
    gap: var(--space-6);
  }
}

/* Desktop styles */
@media (min-width: 1024px) {
  .card-grid {
    grid-template-columns: repeat(3, 1fr);
  }

  .sidebar {
    position: static;
  }
}

/* Large desktop styles */
@media (min-width: 1280px) {
  .container {
    max-width: 1200px;
  }

  .card-grid {
    grid-template-columns: repeat(4, 1fr);
  }
}
```


Responsive Typography

```
/* Fluid typography using clamp() */
.text-display-xl {
  font-size: clamp(2.5rem, 5vw, 4rem);
}

.text-h1 {
  font-size: clamp(1.5rem, 3vw, 2.5rem);
}

.text-h2 {
  font-size: clamp(1.25rem, 2.5vw, 2rem);
}

/* Mobile adjustments */
@media (max-width: 768px) {
  .dashboard-title {
    font-size: var(--font-size-h2);
  }

  .main-content {
    padding: var(--space-4);
  }

  .card-content {
    padding: var(--space-4);
  }
}
```



Animation & Interactions

CSS Variables for Animations

```
:root {
  --transition-duration: 0.3s;
  --transition-timing: cubic-bezier(0.4, 0, 0.2, 1);
  --animation-bounce: cubic-bezier(0.68, -0.55, 0.265, 1.55);
}
```

Keyframe Animations

```
/* Loading Animations */
@keyframes spin {
  from { transform: rotate(0deg); }
  to { transform: rotate(360deg); }
}

@keyframes pulse {
```

```

    0%, 100% { opacity: 1; }
    50% { opacity: 0.5; }
}

@keyframes bounce {
    0%, 20%, 53%, 80%, 100% { transform: translateY(0); }
    40%, 43% { transform: translateY(-8px); }
    70% { transform: translateY(-4px); }
    90% { transform: translateY(-2px); }
}

/* Train Movement Animation */
@keyframes trainMove {
    0% { transform: translateX(-4px); }
    100% { transform: translateX(4px); }
}

/* Gradient Animation */
@keyframes gradientShift {
    0% { background-position: 0% 50%; }
    50% { background-position: 100% 50%; }
    100% { background-position: 0% 50%; }
}

/* Slide Animations */
@keyframes slideInLeft {
    from {
        opacity: 0;
        transform: translateX(-20px);
    }
    to {
        opacity: 1;
        transform: translateX(0);
    }
}

@keyframes slideInUp {
    from {
        opacity: 0;
        transform: translateY(20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

@keyframes fadeIn {
    from { opacity: 0; }
    to { opacity: 1; }
}

@keyframes scaleIn {

```

```
from {
  opacity: 0;
  transform: scale(0.9);
}
to {
  opacity: 1;
  transform: scale(1);
}
}
```

Hover Effects

```
/* Card hover effects */
.interactive-card {
  transition: all var(--transition-duration) var(--transition-timing);
}

.interactive-card:hover {
  transform: translateY(-4px);
  box-shadow: 0 10px 25px rgba(30, 58, 138, 0.15);
}

/* Button hover effects */
.btn {
  transition: all var(--transition-duration) var(--transition-timing);
}

.btn:hover {
  transform: translateY(-1px);
  box-shadow: 0 4px 12px rgba(0, 0, 0, 0.15);
}

.btn:active {
  transform: translateY(0);
}

/* Link hover effects */
.animated-link {
  position: relative;
  text-decoration: none;
  transition: color var(--transition-duration) ease;
}

.animated-link::after {
  content: '';
  position: absolute;
  bottom: -2px;
  left: 0;
  width: 0;
  height: 2px;
  background: var(--primary-blue);
  transition: width var(--transition-duration) ease;
}
```

```
}

.animated-link:hover::after {
  width: 100%;
}
```

Loading States

```
.loading-spinner {
  width: 24px;
  height: 24px;
  border: 2px solid var(--border-color);
  border-top: 2px solid var(--primary-blue);
  border-radius: 50%;
  animation: spin 1s linear infinite;
}

.skeleton {
  background: linear-gradient(90deg, #f0f0f0 25%, #e0e0e0 50%, #f0f0f0 75%);
  background-size: 200% 100%;
  animation: shimmer 1.5s infinite;
  border-radius: var(--border-radius);
}

@keyframes shimmer {
  0% { background-position: -200% 0; }
  100% { background-position: 200% 0; }
}
```

Accessibility Guidelines

WCAG Compliance

RailLynk follows **WCAG 2.1 Level AA** guidelines:

Color Contrast

```
/* High contrast ratios */
:root {
  /* Text on background combinations meet 4.5:1 ratio minimum */
  --contrast-primary: #111827 on #ffffff; /* 16.75:1 */
  --contrast-secondary: #374151 on #ffffff; /* 8.89:1 */
  --contrast-tertiary: #6b7280 on #ffffff; /* 4.69:1 */

  /* Button contrasts meet 3:1 minimum for large text */
  --contrast-btn-primary: #ffffff on #1e40af; /* 8.59:1 */
  --contrast-btn-secondary: #1e40af on #ffffff; /* 8.59:1 */
}
```

Focus States

```
/* Visible focus indicators */
.focusable {
  outline: none;
  position: relative;
}

.focusable:focus-visible::after {
  content: '';
  position: absolute;
  top: -2px;
  left: -2px;
  right: -2px;
  bottom: -2px;
  border: 2px solid var(--primary-blue);
  border-radius: calc(var(--border-radius) + 2px);
  box-shadow: 0 0 0 2px rgba(30, 64, 175, 0.2);
}

/* Skip to content link */
.skip-link {
  position: absolute;
  top: -40px;
  left: 6px;
  background: var(--primary-blue);
  color: var(--text-inverse);
  padding: var(--space-2) var(--space-4);
  border-radius: var(--border-radius);
  text-decoration: none;
  z-index: 3000;
  transition: top var(--transition-duration) ease;
}

.skip-link:focus {
  top: 6px;
}
```

Screen Reader Support

```
/* Screen reader only content */
.sr-only {
  position: absolute;
  width: 1px;
  height: 1px;
  padding: 0;
  margin: -1px;
  overflow: hidden;
  clip: rect(0, 0, 0, 0);
  white-space: nowrap;
  border: 0;
}
```

```
.sr-only-focusable:focus {  
  position: static;  
  width: auto;  
  height: auto;  
  padding: inherit;  
  margin: inherit;  
  overflow: visible;  
  clip: auto;  
  white-space: normal;  
}
```

ARIA Labels and Roles

```
<!-- Navigation landmarks -->  
<nav role="navigation" aria-label="Main navigation">  
  <ul>  
    <li><a href="/" aria-current="page">Home</a></li>  
    <li><a href="/dashboard">Dashboard</a></li>  
  </ul>  
</nav>  
  
<!-- Form labels -->  
<label for="email">Email Address</label>  
<input  
  id="email"  
  type="email"  
  aria-required="true"  
  aria-describedby="email-help"  
  aria-invalid="false"  
>  
<div id="email-help" class="form-help">  
  Enter your registered email address  
</div>  
  
<!-- Button accessibility -->  
<button  
  aria-label="Close modal dialog"  
  aria-pressed="false"  
  type="button"  
>  
  <span aria-hidden="true">&times;</span>  
</button>  
  
<!-- Live regions for dynamic content -->  
<div aria-live="polite" aria-atomic="true" id="status-updates">  
  <!-- Status messages appear here -->  
</div>
```

Shadow System

```
:root {
  --shadow-xs: 0 1px 2px rgba(0, 0, 0, 0.05);
  --shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.1), 0 1px 2px rgba(0, 0, 0, 0.06);
  --shadow-md: 0 4px 6px rgba(0, 0, 0, 0.07), 0 2px 4px rgba(0, 0, 0, 0.06);
  --shadow-lg: 0 10px 15px rgba(0, 0, 0, 0.1), 0 4px 6px rgba(0, 0, 0, 0.05);
  --shadow-xl: 0 20px 25px rgba(0, 0, 0, 0.1), 0 10px 10px rgba(0, 0, 0, 0.04);
  --shadow-2xl: 0 25px 50px rgba(0, 0, 0, 0.25);

  /* Colored shadows */
  --shadow-primary: 0 4px 12px rgba(30, 64, 175, 0.15);
  --shadow-success: 0 4px 12px rgba(16, 185, 129, 0.15);
  --shadow-warning: 0 4px 12px rgba(245, 158, 11, 0.15);
  --shadow-error: 0 4px 12px rgba(239, 68, 68, 0.15);
}
```

Border Radius System

```
:root {
  --border-radius-none: 0;
  --border-radius-sm: 0.25rem;      /* 4px */
  --border-radius: 0.375rem;        /* 6px */
  --border-radius-md: 0.5rem;        /* 8px */
  --border-radius-lg: 0.75rem;       /* 12px */
  --border-radius-xl: 1rem;          /* 16px */
  --border-radius-2xl: 1.5rem;       /* 24px */
  --border-radius-full: 50%;
  --border-radius-pill: 9999px;
}
```

Border System

```
:root {
  --border-width-0: 0;
  --border-width-1: 1px;
  --border-width-2: 2px;
  --border-width-4: 4px;
  --border-width-8: 8px;

  --border-color: #e5e7eb;
  --border-color-light: #f3f4f6;
  --border-color-dark: #d1d5db;
  --border-color-primary: var(--primary-blue);
  --border-color-success: var(--success-color);
  --border-color-warning: var(--warning-color);
  --border-color-error: var(--error-color);
}
```

Z-Index Scale

```
:root {
  --z-index-dropdown: 1000;
  --z-index-sticky: 1010;
  --z-index-fixed: 1020;
  --z-index-modal-backdrop: 1030;
  --z-index-modal: 1040;
  --z-index-popover: 1050;
  --z-index-tooltip: 1060;
  --z-index-toast: 1070;
}
```



User Interface Patterns

Dashboard Patterns

Admin Dashboard

```
.admin-dashboard {
  display: grid;
  grid-template-areas:
    "header header header"
    "sidebar main main"
    "sidebar main main";
  grid-template-columns: 220px 1fr 1fr;
  grid-template-rows: auto 1fr 1fr;
  min-height: 100vh;
  gap: 0;
}

.admin-header {
  grid-area: header;
  background: linear-gradient(135deg, var(--deep-navy), var(--primary-blue));
  color: var(--text-inverse);
  padding: var(--space-4) var(--space-8);
  display: flex;
  align-items: center;
  justify-content: space-between;
}

.admin-sidebar {
  grid-area: sidebar;
  background: var(--deep-navy);
  color: var(--text-inverse);
}

.admin-main {
  grid-area: main;
  background: var(--background-secondary);
}
```



```
padding: var(--space-8);
overflow-y: auto;
}
```

Passenger Dashboard

```
.passenger-dashboard {
  background: linear-gradient(135deg, var(--primary-blue), var(--sky-blue));
  min-height: 100vh;
  color: var(--text-inverse);
}

.passenger-content {
  max-width: 1200px;
  margin: 0 auto;
  padding: var(--space-8);
}

.welcome-section {
  text-align: center;
  margin-bottom: var(--space-12);
}

.balance-display {
  background: rgba(255, 255, 255, 0.1);
  backdrop-filter: blur(10px);
  border-radius: var(--border-radius-xl);
  padding: var(--space-8);
  margin-bottom: var(--space-8);
  border: 1px solid rgba(255, 255, 255, 0.2);
}
```

Form Patterns

Multi-step Form

```
.multi-step-form {
  background: var(--surface-elevated);
  border-radius: var(--border-radius-xl);
  overflow: hidden;
  box-shadow: var(--shadow-xl);
  max-width: 600px;
  margin: 0 auto;
}

.step-indicator {
  display: flex;
  background: var(--background-secondary);
  padding: var(--space-6) var(--space-8);
}

.step {
```

```
flex: 1;
display: flex;
align-items: center;
gap: var(--space-3);
color: var(--text-tertiary);
font-weight: var(--font-weight-medium);
}

.step.active {
  color: var(--primary-blue);
}

.step.completed {
  color: var(--success-color);
}

.step-number {
  width: 32px;
  height: 32px;
  border-radius: 50%;
  display: flex;
  align-items: center;
  justify-content: center;
  font-size: var(--font-size-sm);
  font-weight: var(--font-weight-bold);
  background: var(--background-tertiary);
  border: 2px solid var(--border-color);
}

.step.active .step-number {
  background: var(--primary-blue);
  color: var(--text-inverse);
  border-color: var(--primary-blue);
}

.step.completed .step-number {
  background: var(--success-color);
  color: var(--text-inverse);
  border-color: var(--success-color);
}

.form-step {
  padding: var(--space-8);
}

.form-actions {
  display: flex;
  justify-content: space-between;
  padding: var(--space-6) var(--space-8);
  border-top: 1px solid var(--border-color);
  background: var(--background-secondary);
}
```

Map Integration Patterns

Live Tracking Map

```
.map-container {
  background: var(--surface-elevated);
  border-radius: var(--border-radius-lg);
  overflow: hidden;
  box-shadow: var(--shadow-lg);
  border: 1px solid var(--border-color);
  height: 500px;
  position: relative;
}

.map-controls {
  position: absolute;
  top: var(--space-4);
  right: var(--space-4);
  z-index: 1000;
  display: flex;
  flex-direction: column;
  gap: var(--space-2);
}

.map-control-btn {
  background: var(--surface-elevated);
  border: 1px solid var(--border-color);
  border-radius: var(--border-radius);
  padding: var(--space-3);
  box-shadow: var(--shadow-sm);
  cursor: pointer;
  color: var(--text-secondary);
  transition: all var(--transition-duration) ease;
}

.map-control-btn:hover {
  background: var(--primary-blue);
  color: var(--text-inverse);
  border-color: var(--primary-blue);
}

.train-popup {
  background: var(--surface-elevated);
  border-radius: var(--border-radius);
  padding: var(--space-4);
  box-shadow: var(--shadow-md);
  border: 1px solid var(--border-color);
  max-width: 300px;
}

.train-popup-header {
  display: flex;
```

```

    align-items: center;
    gap: var(--space-2);
    margin-bottom: var(--space-3);
    padding-bottom: var(--space-3);
    border-bottom: 1px solid var(--border-color);
}

.train-icon {
    color: var(--primary-blue);
    animation: trainMove 2s ease-in-out infinite alternate;
}

.train-info {
    display: grid;
    grid-template-columns: auto 1fr;
    gap: var(--space-2) var(--space-4);
    font-size: var(--font-size-sm);
}

.info-label {
    font-weight: var(--font-weight-medium);
    color: var(--text-tertiary);
}

.info-value {
    font-weight: var(--font-weight-semibold);
    color: var(--text-primary);
}

```

Implementation Guidelines

CSS Architecture

RailLynk follows a **modular CSS architecture** with clear organization:

```

src/styles/
├─ index.css                # Global styles and CSS variables
├─ components/              # Component-specific styles
│   └─ navbar.css
│   └─ sidebar.css
│   └─ buttons.css
│   └─ forms.css
│   └─ cards.css
│   └─ tables.css
│   └─ modals.css
├─ pages/                   # Page-specific styles
│   └─ home.css
│   └─ login.css
│   └─ dashboard.css
│   └─ about.css
│   └─ register.css

```

```
├─ utilities/                                # Utility classes
│   ├── spacing.css
│   ├── typography.css
│   ├── colors.css
│   └── responsive.css
└─ animations/                              # Animation definitions
    ├── keyframes.css
    ├── transitions.css
    └── interactions.css
```

Naming Conventions

BEM Methodology

```
/* Block */
.card { }

/* Element */
.card__header { }
.card__content { }
.card__footer { }

/* Modifier */
.card--elevated { }
.card--compact { }
.card__header--primary { }
```

Utility Classes

```
/* Spacing utilities */
.p-0 { padding: 0; }
.p-1 { padding: var(--space-1); }
.p-2 { padding: var(--space-2); }
.mb-4 { margin-bottom: var(--space-4); }

/* Text utilities */
.text-center { text-align: center; }
.text-primary { color: var(--primary-blue); }
.font-bold { font-weight: var(--font-weight-bold); }

/* Display utilities */
.flex { display: flex; }
.grid { display: grid; }
.hidden { display: none; }
.block { display: block; }
```

Performance Considerations

CSS Optimization

```
/* Use CSS custom properties for dynamic values */
.dynamic-element {
  background: hsl(var(--hue), 70%, 50%);
  transform: translateX(var(--offset));
}

/* Prefer transforms over layout-affecting properties */
.animate-element {
  transform: translateY(var(--offset));
  /* Instead of: top: var(--offset); */
}

/* Use will-change for elements that will be animated */
.will-animate {
  will-change: transform, opacity;
}

/* Remove will-change after animation */
.animation-complete {
  will-change: auto;
}
```

Critical CSS

Load critical styles inline and defer non-critical CSS:

```
<head>
  <style>
    /* Critical CSS - above-the-fold styles */
    body { font-family: system-ui; }
    .navbar { background: #1e3a8a; }
  </style>

  <link rel="preload" href="/styles/main.css" as="style"
onload="this.onload=null;this.rel='stylesheet'">
  <noscript><link rel="stylesheet" href="/styles/main.css"></noscript>
</head>
```

Component Documentation Template

```
/**
 * Card Component
 *
 * A flexible content container with optional header and footer.
 *
 * @example
 * <div class="card card--elevated">
```

```

*   <header class="card__header">
*     <h3>Card Title</h3>
*   </header>
*   <div class="card__content">
*     Card content goes here
*   </div>
*   <footer class="card__footer">
*     <button class="btn btn--primary">Action</button>
*   </footer>
* </div>
*
* @modifiers
* --elevated      Adds elevated shadow
* --compact       Reduces padding
* --borderless    Removes border
*
* @since 2.0.0
*/
.card {
  background: var(--surface-elevated);
  border: 1px solid var(--border-color);
  border-radius: var(--border-radius-lg);
  overflow: hidden;
}

```

Quality Assurance

Browser Compatibility

RailLynk supports:

- **Chrome** 90+
- **Firefox** 88+
- **Safari** 14+
- **Edge** 90+

Testing Checklist

Visual Testing

- ☐ All colors meet WCAG contrast requirements
- ☐ Typography scales appropriately across devices
- ☐ Interactive elements have clear hover/focus states
- ☐ Animations are smooth and purposeful
- ☐ Loading states are implemented
- ☐ Error states are clearly visible

Responsive Testing

- ☐ Mobile devices (320px - 768px)
- ☐ Tablets (768px - 1024px)
- ☐ Desktop (1024px+)
- ☐ Large screens (1400px+)

Accessibility Testing

- ☐ Keyboard navigation works properly
- ☐ Screen readers can access all content
- ☐ Focus indicators are visible
- ☐ Color is not the only way to convey information
- ☐ Text can be resized to 200% without horizontal scrolling

Performance Testing

- ☐ CSS file sizes are optimized
- ☐ Critical CSS is inlined
- ☐ Animations use GPU acceleration
- ☐ No layout shift during loading
- ☐ Images are optimized and responsive

Design System Maintenance

Version Control

- Track all design system changes in Git
- Use semantic versioning (2.0.0, 2.1.0, etc.)
- Document breaking changes in CHANGELOG.md
- Create migration guides for major updates

Review Process

1. **Design Review:** UI/UX team approves visual changes
2. **Code Review:** Development team reviews implementation
3. **Accessibility Review:** Check WCAG compliance
4. **Testing:** Cross-browser and device testing
5. **Documentation:** Update design manual and component docs

Monitoring & Analytics

- Track component usage across the application
- Monitor performance metrics
- Gather user feedback on UI/UX

		handlers.js	# MSW request handlers
		server.js	# MSW server setup
		setupTests.js	# Global test configuration

Test Categories & Examples

1. Unit Testing

Purpose: Test individual components in isolation

Real Example from AdminDashboard.test.jsx:

```
import { describe, it, beforeEach, afterEach, expect, vi, beforeAll } from "vitest";
import { render, screen, waitFor, fireEvent } from "@testing-library/react";
import { MemoryRouter } from "react-router-dom";
import AdminDashboard from "../../components/AdminDashboard";

// Mock recharts components
vi.mock('recharts', () => ({
  LineChart: ({ children }) => <div data-testid="line-chart">{children}</div>,
  Bar: () => <div data-testid="bar" />,
  PieChart: ({ children }) => <div data-testid="pie-chart">{children}</div>,
  ResponsiveContainer: ({ children }) => <div data-testid="responsive-container">{children}</div>,
}));

// Mock ResizeObserver for chart components
beforeAll(() => {
  global.ResizeObserver = class {
    observe() {}
    unobserve() {}
    disconnect() {}
  };
});

// Mock API calls
vi.mock("../../api", () => ({
  default: {
    get: vi.fn(),
  },
}));

describe("AdminDashboard Component", () => {
  it("renders loading state initially", () => {
    render(<MemoryRouter><AdminDashboard /></MemoryRouter>);
    expect(screen.getByText("Loading...")).toBeInTheDocument();
  });

  it("displays dashboard stats after loading", async () => {
    API.get.mockResolvedValue({
      data: {
        total_cards: 8,
        total_passengers: 150,
      },
    });
  });
});
```

```

        total_stations: 45,
        daily_revenue: 2500,
        monthly_revenue: 25000
    }
});

render(<MemoryRouter><AdminDashboard /></MemoryRouter>);

await waitFor(() => {
    expect(screen.getByText("8")).toBeInTheDocument();
    expect(screen.getByText("150")).toBeInTheDocument();
});
});
});

```

2. Integration Testing

Purpose: Test component interactions and complete user workflows

Real Example from Login.integration.test.jsx:

```

import { render, screen, fireEvent, waitFor } from "@testing-library/react";
import Login from "../../pages/Login";
import { BrowserRouter } from "react-router-dom";
import { vi } from "vitest";
import API from "../../api";

// Mock localStorage
const localStorageMock = (() => {
    let store = {};
    return {
        getItem: (key) => store[key],
        setItem: (key, value) => (store[key] = value),
        clear: () => (store = {}),
    };
})();
Object.defineProperty(window, "localStorage", { value: localStorageMock });

// Mock API
vi.mock("../../api", () => ({
    default: {
        post: vi.fn(),
    },
}));

describe("Login Integration Test", () => {
    it("logs in a passenger and redirects to /passengerDashboard", async () => {
        API.post.mockResolvedValue({
            data: {
                access: "access_token",
                refresh: "refresh_token",
                user_type: "passenger",
                profile: { nic_number: "123456789V" },
            },
        });
    });
});

```

```

    },
  });

  render(<BrowserRouter><Login /></BrowserRouter>);

  // Fill form
  fireEvent.change(screen.getByLabelText(/username/i), {
    target: { value: "passenger@test.com" },
  });
  fireEvent.change(screen.getByLabelText(/password/i), {
    target: { value: "password123" },
  });

  // Submit form
  fireEvent.click(screen.getByRole("button", { name: /login/i }));

  // Verify API call
  await waitFor(() => {
    expect(API.post).toHaveBeenCalledWith("/auth/login", {
      username: "passenger@test.com",
      password: "password123",
    });
  });
});
});
});

```

3. Component-Specific Testing Patterns

ProtectedRoute Testing:

```

describe("ProtectedRoute Component", () => {
  it("renders protected content when user has correct role", () => {
    localStorage.setItem("userType", "admin");

    render(
      <BrowserRouter>
        <ProtectedRoute allowedRoles={["admin"]}>
          <div>Protected Admin Content</div>
        </ProtectedRoute>
      </BrowserRouter>
    );

    expect(screen.getByText("Protected Admin Content")).toBeInTheDocument();
  });

  it("redirects when user lacks required role", () => {
    localStorage.setItem("userType", "passenger");

    render(
      <BrowserRouter>
        <ProtectedRoute allowedRoles={["admin"]}>
          <div>Protected Admin Content</div>
        </ProtectedRoute>
      </BrowserRouter>
    );
  });
});

```

```

    </BrowserRouter>
  );

  expect(screen.queryByText("Protected Admin Content")).not.toBeInTheDocument();
});
});

```

Mock Strategies

API Mocking with MSW

Current Implementation in handlers.js:

```

import { rest } from 'msw';

export const handlers = [
  // Registration endpoint
  rest.post('/api/register', (req, res, ctx) => {
    return res(
      ctx.status(200),
      ctx.json({ message: 'Registration successful' })
    );
  }),

  // Add more handlers for different endpoints
  rest.get('/api/admin/dashboard', (req, res, ctx) => {
    return res(
      ctx.json({
        total_cards: 12,
        total_passengers: 120,
        total_stations: 50,
        daily_revenue: 4500,
        monthly_revenue: 12000,
        last_5_months_revenue: {
          "May": 12000,
          "Apr": 11000,
          "Mar": 9000,
          "Feb": 10000,
          "Jan": 8000
        },
        most_busy_stations_today: [
          { station_name: 'Colombo Fort', usage_count: 70 },
          { station_name: 'Kandy', usage_count: 60 },
          { station_name: 'Galle', usage_count: 50 }
        ]
      })
    );
  }),
];

```

Server Setup in server.js:

```
import { setupServer } from 'msw/node';
import { handlers } from './handlers';

export const server = setupServer(...handlers);

beforeAll(() => server.listen());
afterEach(() => server.resetHandlers());
afterAll(() => server.close());
```

Component Mocking

Chart Components (Recharts):

```
vi.mock('recharts', () => ({
  LineChart: ({ children }) => <div data-testid="line-chart">{children}</div>,
  BarChart: ({ children }) => <div data-testid="bar-chart">{children}</div>,
  PieChart: ({ children }) => <div data-testid="pie-chart">{children}</div>,
  ResponsiveContainer: ({ children }) => <div data-testid="responsive-container">{children}</div>,
  XAxis: () => <div data-testid="x-axis" />,
  YAxis: () => <div data-testid="y-axis" />,
  CartesianGrid: () => <div data-testid="cartesian-grid" />,
  Tooltip: () => <div data-testid="tooltip" />,
  Legend: () => <div data-testid="legend" />,
}));
```

Browser APIs:

```
// localStorage mock
const localStorageMock = (() => {
  let store = {};
  return {
    getItem: (key) => store[key],
    setItem: (key, value) => (store[key] = value),
    clear: () => (store = {}),
  };
})();
Object.defineProperty(window, "localStorage", { value: localStorageMock });

// ResizeObserver mock for charts
beforeAll(() => {
  global.ResizeObserver = class {
    observe() {}
    unobserve() {}
    disconnect() {}
  };
});
```

Test Configuration

Vitest Configuration (vite.config.js)

```
export default defineConfig({
  plugins: [react()],
  test: {
    globals: true,
    environment: "jsdom",
    setupFiles: "./src/setupTests.js",
  },
});
```

Test Setup (setupTests.js)

```
import '@testing-library/jest-dom';
```

Testing Utilities & Helpers

Router Wrapper

```
const renderWithRouter = (ui, { initialEntries = ['/'] } = {}) => {
  return render(
    <MemoryRouter initialEntries={initialEntries}>
      {ui}
    </MemoryRouter>
  );
};

// Usage in tests
renderWithRouter(<Login />);
```

Form Testing Utilities

```
const fillLoginForm = (email, password) => {
  fireEvent.change(screen.getByLabelText(/username/i), {
    target: { value: email },
  });
  fireEvent.change(screen.getByLabelText(/password/i), {
    target: { value: password },
  });
};

const submitForm = () => {
  fireEvent.click(screen.getByRole("button", { name: /login/i }));
};
```

Component Testing Patterns

Dashboard Components Testing

```
// Test loading states
it("renders loading state initially", () => {
  render(<MemoryRouter><AdminDashboard /></MemoryRouter>);
  expect(screen.getByText("Loading...")).toBeInTheDocument();
});

// Test data display
it("displays dashboard statistics", async () => {
  API.get.mockResolvedValue({
    data: mockDashboardData
  });

  render(<MemoryRouter><AdminDashboard /></MemoryRouter>);

  await waitFor(() => {
    expect(screen.getByText("Total Cards")).toBeInTheDocument();
    expect(screen.getByText("150")).toBeInTheDocument(); // passenger count
  });
});
```

Form Components Testing

```
// Test form rendering
it("renders registration form fields", () => {
  render(<BrowserRouter><PassengerRegister /></BrowserRouter>);

  expect(screen.getByLabelText(/first name/i)).toBeInTheDocument();
  expect(screen.getByLabelText(/email/i)).toBeInTheDocument();
  expect(screen.getByLabelText(/nic number/i)).toBeInTheDocument();
});

// Test form validation
it("shows validation errors for invalid inputs", async () => {
  render(<BrowserRouter><PassengerRegister /></BrowserRouter>);

  fireEvent.click(screen.getByRole("button", { name: /register/i }));

  await waitFor(() => {
    expect(screen.getByText(/email is required/i)).toBeInTheDocument();
  });
});
```


Navigation Components Testing

```
// Test conditional rendering based on user type
it("shows admin navigation for admin users", () => {
  localStorage.setItem("userType", "admin");

  render(<BrowserRouter><Navbar /></BrowserRouter>);

  expect(screen.getByText("Admin Dashboard")).toBeInTheDocument();
  expect(screen.queryByText("Passenger Dashboard")).not.toBeInTheDocument();
});
```

Running Tests

Available Test Commands

```
# Run all tests
npm test

# Run tests in watch mode (development)
npm test -- --watch

# Run tests with coverage report
npm test -- --coverage

# Run specific test file
npm test -- AdminDashboard.test.jsx

# Run tests with specific pattern
npm test -- --testNamePattern="should render"

# Run only unit tests
npm test -- src/__test__/unit/

# Run only integration tests
npm test -- src/__test__/integration/

# Run tests in silent mode
npm test -- --silent

# Run with verbose output
npm test -- --reporter=verbose
```

Test Execution Results

- ✅ Test Files: 29 passed (29)
- ✅ Tests: 103 passed (103)
- 🕒 Duration: ~13-15 seconds
- 📊 Coverage: Available via --coverage flag
- 🔄 Watch Mode: Available for development

Performance Testing

Component Rendering Performance

```
it("renders large data sets efficiently", async () => {
  const largeDataset = Array.from({ length: 1000 }, (_, i) => ({
    id: i,
    name: `Station ${i}`,
    usage_count: Math.floor(Math.random() * 100)
  }));

  API.get.mockResolvedValue({ data: largeDataset });

  const start = performance.now();
  render(<MemoryRouter><StationList /></MemoryRouter>);

  await waitFor(() => {
    expect(screen.getByText("Station 0")).toBeInTheDocument();
  });

  const end = performance.now();
  expect(end - start).toBeLessThan(2000); // Should render within 2 seconds
});
```

Accessibility Testing

Current Accessibility Patterns

```
// Test proper labeling
it("has accessible form labels", () => {
  render(<BrowserRouter><Login /></BrowserRouter>);

  expect(screen.getByLabelText(/username/i)).toBeInTheDocument();
  expect(screen.getByLabelText(/password/i)).toBeInTheDocument();
});

// Test keyboard navigation
it("supports keyboard navigation", () => {
  render(<BrowserRouter><Navbar /></BrowserRouter>);

  const firstLink = screen.getByRole("link", { name: /home/i });
  firstLink.focus();

  expect(firstLink).toHaveFocus();
});
```

Error Handling Testing

API Error Testing

```
it("handles API errors gracefully", async () => {
  API.post.mockRejectedValue({
    response: { data: { error: 'Network error' } }
  });

  render(<BrowserRouter><Login /></BrowserRouter>);

  fillLoginForm("test@example.com", "password");
  submitForm();

  await waitFor(() => {
    expect(screen.getByText(/network error/i)).toBeInTheDocument();
  });
});
```

Test Maintenance & Best Practices

Current Test Quality Metrics

- **Test Coverage:** Comprehensive coverage across all major components
- **Test Speed:** Average 13.74s for full suite execution
- **Test Reliability:** 100% pass rate (103/103 tests passing)
- **Maintainability:** Well-organized test structure with clear naming

Testing Best Practices Used

1. **Isolation:** Each test is independent with proper cleanup
2. **Mocking:** External dependencies are properly mocked
3. **Assertions:** Clear and specific assertions
4. **Organization:** Logical grouping of related tests
5. **Documentation:** Descriptive test names and comments

Continuous Integration Ready

```
# Example GitHub Actions workflow
name: RailLynk Tests
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3
      - uses: actions/setup-node@v3
        with:
          node-version: '18'
```

```
- run: npm ci
- run: npm test -- --run
- run: npm test -- --coverage
```

Debugging Tests

Test Debugging Techniques








```
// Use screen.debug() to see DOM output
it("debugs component rendering", () => {
  render(<MyComponent />);
  screen.debug(); // Prints DOM to console
});

// Debug specific elements
it("debugs specific element", () => {
  render(<MyComponent />);
  const button = screen.getByRole('button');
  console.log(button.outerHTML);
});






// Debug failed queries
it("debugs queries", () => {
  render(<MyComponent />);
  screen.getByText('Non-existent text'); // Shows helpful error message
});
```

Current Test Suite Coverage

Components with Comprehensive Tests

-  **AdminDashboard:** 5 tests (loading, rendering, API integration)
-  **Login:** 4 unit tests + 2 integration tests
-  **ProtectedRoute:** 8 tests (authentication scenarios)
-  **PassengerDashboard:** Complete rendering tests
-  **AnnouncementCard:** 13 comprehensive tests
-  **AdminLogin:** 15 detailed tests
-  **Forms:** Registration and transaction forms tested

Integration Test Coverage

-  **Login Flow:** Complete user authentication workflow
-  **Registration:** Passenger and station registration
-  **Dashboard:** Admin dashboard data loading
-  **Transactions:** Passenger transaction page functionality
-  **Recharge:** Card recharge workflow

Extending the Test Suite

Adding New Tests

```
// Template for new component tests
import { describe, it, expect, vi } from 'vitest';
import { render, screen, waitFor } from '@testing-library/react';
import { BrowserRouter } from 'react-router-dom';
import YourComponent from '../components/YourComponent';

describe('YourComponent', () => {
  it('renders correctly', () => {
    render(
      <BrowserRouter>
        <YourComponent />
      </BrowserRouter>
    );

    expect(screen.getByRole('button')).toBeInTheDocument();
  });

  it('handles user interactions', async () => {
    render(
      <BrowserRouter>
        <YourComponent />
      </BrowserRouter>
    );

    fireEvent.click(screen.getByRole('button'));

    await waitFor(() => {
      expect(screen.getByText('Expected Result')).toBeInTheDocument();
    });
  });
});
```

Resources & References

Design Inspiration

- [Railway industry design patterns](#)
- [Material Design principles](#)
- [Airbnb Design System](#)

Tools & Libraries

- **CSS Framework:** Custom CSS with utility classes
- **Icons:** Font Awesome, React Icons
- **Animations:** CSS Animations, Framer Motion (if needed)
- **Testing:** Lighthouse, axe-core, Chrome DevTools








Documentation

- [MDN Web Docs](#)
 - [WCAG Guidelines](#)
 - [CSS Grid Guide](#)
 - [Flexbox Guide](#)
-







Updates & Changelog

Updates & Changelog

Version 2.1 - Final (July 22, 2025)

-  **Design Manual Finalized** - Comprehensive analysis of actual codebase
-  **Real Color System** - Updated with actual CSS variables and color usage
-  **Accurate Typography** - Poppins font stack from Google Fonts implementation
-  **React Icons Integration** - Documented actual icon usage patterns (5.5.0)
-  **Layout System Analysis** - Flex/Grid patterns from real components
-  **Component Library Documentation** - Based on existing 19 components
-  **Testing Manual Integration** - Complete test suite documentation (103 tests)

Version 2.0 - Enhanced (July 22, 2025)

-  Complete design system overhaul
-  New component library with 50+ components
-  Enhanced accessibility features
-  Mobile-first responsive design
-  Performance optimizations
-  Animation system implementation

Actual Implementation Analysis

Codebase Statistics:

- **Components:** 19 React components analyzed
- **Styles:** 18 CSS files examined

- **Pages:** 15 page components documented
- **Test Coverage:** 103 tests across 29 files
- **Dependencies:** React 19.0.0, Vite 6.1.0, Vitest 3.0.5

Key Findings:

- **Primary Color:** `rgb(16, 43, 107)` consistently used across navigation
- **Font System:** Poppins from Google Fonts with Segoe UI fallback
- **Icon Library:** React Icons (Feather collection) exclusively used
- **Layout Pattern:** Fixed sidebar with collapsible functionality
- **Background:** `#f5f7fb` light blue-gray for main application

Architecture Patterns:

- **Component Structure:** Functional components with hooks
- **State Management:** `useState` and `useEffect` for local state
- **Routing:** React Router Dom v7.2.0 with protected routes
- **Styling:** CSS Modules with component-specific stylesheets
- **API Integration:** Axios with interceptors for JWT authentication



Resources & References

Project Resources

- **Repository:** github.com/sandalu-umayanga/3yp-railLynk-front
- **API Base:** <https://raillynk.site/api/>
- **Design System:** This comprehensive manual (2000+ lines)
- **Test Suite:** 103 tests with 100% pass rate

External Resources

- **React Icons:** react-icons.github.io/react-icons
- **Google Fonts:** fonts.google.com/specimen/Poppins
- **Recharts:** recharts.org - Used for dashboard visualizations
- **React Leaflet:** react-leaflet.js.org - Map integration

Tools & Technologies

- **Framework:** React 19.0.0 with Vite 6.1.0
- **Testing:** Vitest 3.0.5 + React Testing Library 16.2.0
- **Icons:** React Icons 5.5.0 (Feather collection)
- **Maps:** Leaflet 1.9.4 with React Leaflet 5.0.0
- **Charts:** Recharts 2.15.4 for data visualization

- **HTTP:** Axios 1.7.9 with JWT authentication
- **Routing:** React Router Dom 7.2.0

Documentation References

- [MDN Web Docs](#) - CSS and JavaScript reference
- [WCAG Guidelines](#) - Accessibility standards
- [React Documentation](#) - React 19 features and patterns
- [Vite Guide](#) - Build tool configuration

Design Inspiration Sources

- **Railway Industry:** Real-world railway management interfaces
- **Modern Dashboards:** Clean, professional admin interfaces
- **Mobile-First:** Responsive design patterns for transportation apps
- **Accessibility:** Universal design principles for public services



Implementation Notes

Getting Started with RailLynk Design System

1. **Clone Repository:** `git clone https://github.com/sandalu-umayanga/3yp-railLynk-front`
2. **Install Dependencies:** `npm install`
3. **Start Development:** `npm run dev`
4. **Run Tests:** `npm test`
5. **Build Production:** `npm run build`

Key Configuration Files

- **Vite Config:** `vite.config.js` - Build and test configuration
- **Package.json:** Dependencies and scripts
- **Index.css:** Global CSS variables and base styles
- **setupTests.js:** Test environment configuration

Development Workflow

1. **Component Creation:** Follow established patterns in `/src/components/`
2. **Styling:** Component-specific CSS in `/src/styles/`
3. **Testing:** Unit tests in `/src/__test__/unit/`, integration tests in `/src/__test__/integration/`
4. **API Integration:** Use `/src/api.js` for HTTP requests
5. **Routing:** Protected routes pattern with role-based access

Best Practices Followed

- **Consistent Naming:** PascalCase components, kebab-case CSS files
 - **Modular CSS:** Component-specific stylesheets
 - **Responsive Design:** Mobile-first approach with breakpoints
 - **Accessibility:** Semantic HTML, ARIA labels, keyboard navigation
 - **Performance:** Code splitting, lazy loading, optimized assets
 - **Testing:** Comprehensive unit and integration test coverage
-

Design Manual Version: 2.1 Final

Last Updated: July 22, 2025

Total Lines: 2000+

Coverage: Complete RailLynk Frontend Design System