

# THE BITCOIN CORPORATION

## Implementation Guide

*Step-by-Step Development Procedures*

Document Version: 1.0 | Date: November 2025 | The Bitcoin Corporation LTD

---

### Bitcoin OS Design System Implementation Guide

---

#### Overview

This guide demonstrates how to implement the standardized Bitcoin OS design system across all applications while maintaining existing functionality.

#### Standard Layout Implementation

##### 1. Layout Structure

Every Bitcoin app should follow this layout hierarchy:

```
StandardPocBar (40px height, z-index: 9999)
StandardTaskbar (32px height, z-index: 9998, top: 40px)
Main Content (margin-top: 72px to account for fixed headers)
```

##### 2. Implementation Example

#### Import the standardized components:

```
typescript
import StandardPocBar from '../components/StandardPocBar';
import StandardTaskbar from '../components/StandardTaskbar';
import '../styles/theme.css'; // App-specific color theme
```

#### Layout wrapper component:

```
typescript

interface AppLayoutProps {
  children: React.ReactNode;
  appName: string;
  primaryColor: string;
  secondaryColor: string;
  isAuthenticated?: boolean;
  currentUser?: any;
  onLogout?: () => void;
  customMenus?: DropdownMenu[];
}

export default function AppLayout({
  children,
  appName,
  primaryColor,
  secondaryColor,
  isAuthenticated,
  currentUser,
  onLogout,
  customMenus = []
}: AppLayoutProps) {
  return (
    <>
    
      appName={appName}
      primaryColor={primaryColor}
      secondaryColor={secondaryColor}
      links={[
        { label: 'Documentation', href: '/docs' },
        { label: 'Support', href: 'https://support.bitcoinos.com', external: true }
      ]}
    
  )
}
```

```
/>

appName={appName}

primaryColor={primaryColor}

secondaryColor={secondaryColor}

isAuthenticated={isAuthenticated}

currentUser={currentUser}

onLogout={onLogout}

customMenus={customMenus}

/>

{children}

);

}
```

### 3. App-Specific Implementation

#### For each app, update the main layout/page:

```
typescript

// bitcoin-writer/app/page.tsx

export default function Home() {

// ... existing state and logic ...

const writerMenus = [

{

label: 'Document',

items: [

{ label: 'New Document', action: onNewDocument, shortcut: '%N', icon: FileText },

{ label: 'Save Document', action: onSaveDocument, shortcut: '%S', icon: Save },

{ divider: true },

{ label: 'Export', action: onExport, icon: Download }

]

},


{
```

```
label: 'Blockchain',  
      items: [  
        { label: 'Tokenize', action: onOpenTokenizeModal, icon: Coins },  
        { label: 'Publish', action: onPublish, icon: Upload }  
      ]  
    }  
  ];  
  
  return (  
    appName="Bitcoin Writer"  
    primaryColor="#1e40af"  
    secondaryColor="#06d6ff"  
    isAuthenticated={isAuthenticated}  
    currentUser={currentUser}  
    onLogout={onLogout}  
    customMenus={writerMenus}  
  >  
  {/ Existing app content /}  
  {/ All existing components /}  
);  
}
```

## Color Scheme Integration

### 1. Import Theme CSS

Each app should import its theme.css file that defines the color custom properties:

```
css  
/ styles/theme.css /  
:root {  
  --app-primary: #1e40af;  
  --app-secondary: #06d6ff;
```

```
--app-gradient: linear-gradient(135deg, var(--app-primary) 0%, var(--app-secondary) 100%);  
/ ... other color definitions /  
}
```

## 2. Update Existing Components

Replace hard-coded colors with CSS custom properties:

```
css  
/ Before /  
.my-button {  
background: #3b82f6;  
color: white;  
}  
/ After /  
.my-button {  
background: var(--app-primary);  
color: white;  
box-shadow: var(--app-shadow);  
}  
.my-gradient-element {  
background: var(--app-gradient);  
}
```

## Migration Strategy

### Phase 1: Core Apps (`bitcoin-writer`, `bitcoin-email`, `bitcoin-music`)

1. Create theme.css files with app-specific colors
2. Add StandardPocBar and StandardTaskbar components
3. Update layouts to use new structure

4. Test functionality and adjust

## Phase 2: Next.js Apps

1. Apply same pattern to other Next.js apps
2. Ensure consistent menu structures
3. Standardize common functionality

## Phase 3: React to Next.js Conversion

For apps that need React → Next.js conversion:

1. Create new Next.js structure
2. Port existing components
3. Apply standardized design system
4. Test and deploy

## Component Customization

### Custom PoC Bar Links

```
typescript
appName="Bitcoin Music"
primaryColor="#6b21a8"
secondaryColor="#ff1493"

links={[
  { label: 'Artists', href: '/artists' },
  { label: 'Marketplace', href: '/marketplace' },
  { label: 'NFT Guide', href: '/nft-guide' }
]}
/>
```

### Custom Taskbar Menus

```
typescript

const musicMenus = [
  {
    label: 'Library',
    items: [
      { label: 'My Music', href: '/library', icon: Music },
      { label: 'Playlists', href: '/playlists', icon: List },
      { label: 'Upload Track', action: onUpload, icon: Upload }
    ]
  },
  {
    label: 'NFT',
    items: [
      { label: 'Mint NFT', action: onMintNFT, icon: Coins },
      { label: 'My NFTs', href: '/nfts', icon: Image }
    ]
  }
];
```

## Best Practices

### 1. Responsive Design

- Use CSS Grid/Flexbox for layouts
- Ensure components work on mobile devices
- Test taskbar menu behavior on different screen sizes

### 2. Accessibility

- Maintain keyboard navigation
- Ensure color contrasts meet WCAG standards
- Provide proper ARIA labels

### **3. Performance**

- Lazy load heavy components
- Optimize gradient rendering
- Use efficient state management

### **4. Testing**

- Test PoC bar dismiss functionality
- Verify taskbar menus work correctly
- Ensure app-specific functionality is preserved

## **Rollout Schedule**

### **Week 1: Core Infrastructure**

- [ ] Complete standardized components
- [ ] Create theme files for all apps
- [ ] Update main Dock component

### **Week 2: Primary Apps**

- [ ] Implement in bitcoin-writer
- [ ] Implement in bitcoin-email
- [ ] Implement in bitcoin-music

### **Week 3: Secondary Apps**

- [ ] bitcoin-video, bitcoin-photos, bitcoin-art
- [ ] bitcoin-calendar, bitcoin-drive, bitcoin-search

### **Week 4: Remaining Apps & Polish**

- [ ] All remaining apps
- [ ] Cross-app testing

- [ ] Performance optimization
- [ ] Documentation updates

## Quality Assurance

### Checklist for Each App

- [ ] StandardPocBar displays correctly with app colors
- [ ] StandardTaskbar functions with app-specific menus
- [ ] Color scheme applied consistently
- [ ] Existing functionality preserved
- [ ] Mobile responsiveness maintained
- [ ] Performance impact minimal
- [ ] Accessibility standards met

---

*This implementation maintains the unique functionality of each app while providing a consistent, professional appearance across the entire Bitcoin OS ecosystem.*