**LEGAL BUNDLE UI ANALYSIS- Combined Analysis for (Privacy Policy, Terms of Service, Risk Disclosure and Shariah Compliant**

**📋 LEGAL BUNDLE UI ANALYSIS**

***Complete Technical Specifications for Cursor Implementation***

**1. UI Components & Layout**

**Component Architecture**

// Legal Bundle Component Structure

src/pages/legal/

├── PrivacyPolicy.tsx ✅

├── TermsOfService.tsx ✅

├── RiskDisclosure.tsx ✅

├── ShariahCompliance.tsx ✅

├── GDPRCompliance.tsx ✅

├── CookieNotice.tsx ✅

└── LegalLayout.tsx (Recommended abstraction)

**Shared Component Pattern**

// Recommended: Create reusable LegalPageWrapper

interface LegalPageProps {

title: string;

icon: React.ComponentType<{ className?: string }>;

iconColor: string;

children: React.ReactNode;

lastUpdated?: string;

footerBanner?: {

color: string;

message: string;

};

}

const LegalPageWrapper: React.FC<LegalPageProps> = ({

title, icon: Icon, iconColor, children, lastUpdated, footerBanner

}) => {

const { language } = useLanguage();

return (

<Layout>

<div className={`max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 py-12 ${language === 'ar' ? 'rtl' : 'ltr'}`}>

<div className="mb-8 text-center">

<h1 className="text-3xl font-bold text-white mb-4">{title}</h1>

<div className="flex justify-center">

<Icon className={`h-12 w-12 ${iconColor}`} />

</div>

</div>

{children}

{lastUpdated && (

<div className="mt-10 p-4 bg-slate-800/50 border border-slate-700 rounded-lg">

<p className="text-slate-400 text-sm text-center">

Last updated: {lastUpdated}

</p>

</div>

)}

{footerBanner && (

<div className={`mt-10 p-4 bg-${footerBanner.color}-900/20 border border-${footerBanner.color}-800/50 rounded-lg`}>

<p className={`text-${footerBanner.color}-400 text-sm text-center`}>

{footerBanner.message}

</p>

</div>

)}

</div>

</Layout>

);

};

**Interactive Elements**

* **Navigation Links**: Footer links in Layout.tsx and LandingPage.tsx
* **Language Toggle**: Affects RTL/LTR layout and content
* **Scroll-to-Top**: Footer link handlers with smooth scrolling
* **External Links**: Support links (Telegram, email contacts)

**Responsive Design Specifications**

/\* Mobile-first approach consistent across all legal pages \*/

.legal-container {

@apply max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 py-12;

}

.legal-title {

@apply text-3xl font-bold text-white mb-4;

}

.legal-icon {

@apply h-12 w-12;

}

.legal-section {

@apply space-y-6 text-slate-300;

}

.legal-subsection {

@apply text-xl font-semibold text-white mb-4;

}

**2. State Management (Zustand)**

**Legal Pages State Store**

// stores/legalStore.ts

interface LegalState {

// Consent tracking

cookieConsent: boolean | null;

gdprAccepted: boolean | null;

riskDisclosureAccepted: boolean | null;

// User preferences

preferredLegalLanguage: 'en' | 'de' | 'ar';

lastViewedLegalPages: string[];

legalUpdatesViewed: Record<string, string>; // page -> lastViewedVersion

// Actions

updateConsent: (type: 'cookie' | 'gdpr' | 'risk', accepted: boolean) => void;

markLegalPageViewed: (pageName: string) => void;

updateLegalLanguagePreference: (lang: 'en' | 'de' | 'ar') => void;

// Computed

hasRequiredConsents: () => boolean;

needsLegalUpdatesReview: () => string[];

}

const useLegalStore = create<LegalState>((set, get) => ({

cookieConsent: null,

gdprAccepted: null,

riskDisclosureAccepted: null,

preferredLegalLanguage: 'en',

lastViewedLegalPages: [],

legalUpdatesViewed: {},

updateConsent: (type, accepted) => set((state) => ({

[`${type}Accepted`]: accepted,

[`${type}AcceptedAt`]: new Date().toISOString()

})),

markLegalPageViewed: (pageName) => set((state) => ({

lastViewedLegalPages: [...state.lastViewedLegalPages.filter(p => p !== pageName), pageName].slice(-5),

legalUpdatesViewed: {

...state.legalUpdatesViewed,

[pageName]: new Date().toISOString()

}

})),

hasRequiredConsents: () => {

const state = get();

return state.cookieConsent === true && state.gdprAccepted === true;

}

}));

**Local State Patterns**

// Individual legal pages only need minimal local state

const [isScrolled, setIsScrolled] = useState(false);

const [activeSection, setActiveSection] = useState<string | null>(null);

// Intersection Observer for section highlighting

useEffect(() => {

const observer = new IntersectionObserver(

(entries) => {

entries.forEach((entry) => {

if (entry.isIntersecting) {

setActiveSection(entry.target.id);

}

});

},

{ threshold: 0.6 }

);

// Observe all section headers

document.querySelectorAll('h2[id]').forEach((section) => {

observer.observe(section);

});

return () => observer.disconnect();

}, []);

**3. API Contracts & Integration**

**Legal Content Management API**

// Legal pages are primarily static, but may need:

// GET /api/legal/content/:page/:language

interface LegalContentResponse {

page: 'privacy' | 'terms' | 'risk' | 'shariah' | 'gdpr' | 'cookie';

language: 'en' | 'de' | 'ar';

content: {

title: string;

sections: Array<{

id: string;

title: string;

content: string;

subsections?: Array<{

title: string;

content: string;

}>;

}>;

};

lastUpdated: string;

version: string;

}

// POST /api/legal/consent

interface ConsentRequest {

userId?: string;

sessionId: string;

consentType: 'cookie' | 'gdpr' | 'risk';

accepted: boolean;

timestamp: string;

ipAddress: string;

userAgent: string;

}

// GET /api/legal/consent/:sessionId

interface ConsentResponse {

consents: Array<{

type: 'cookie' | 'gdpr' | 'risk';

accepted: boolean;

timestamp: string;

}>;

}

**Legal Compliance Tracking**

// POST /api/analytics/legal-page-view

interface LegalPageViewRequest {

page: string;

language: string;

userId?: string;

sessionId: string;

referrer?: string;

timeSpent?: number;

}

**4. Performance & Optimization**

**Bundle Splitting Strategy**

// Lazy load legal pages since they're not frequently accessed

const PrivacyPolicy = lazy(() => import('../pages/legal/PrivacyPolicy'));

const TermsOfService = lazy(() => import('../pages/legal/TermsOfService'));

const RiskDisclosure = lazy(() => import('../pages/legal/RiskDisclosure'));

const ShariahCompliance = lazy(() => import('../pages/legal/ShariahCompliance'));

const GDPRCompliance = lazy(() => import('../pages/legal/GDPRCompliance'));

const CookieNotice = lazy(() => import('../pages/legal/CookieNotice'));

// Route configuration with Suspense

<Route path="/privacy-policy" element={

<Suspense fallback={<LegalPageSkeleton />}>

<PrivacyPolicy />

</Suspense>

} />

**Content Optimization**

// Memoize static legal content

const LegalContent = React.memo(({ content }: { content: string }) => {

return <div dangerouslySetInnerHTML={{ \_\_html: content }} />;

});

// Optimize language switching

const useMemoizedLegalContent = (content: Record<string, string>, language: string) => {

return useMemo(() => content[language] || content.en, [content, language]);

};

**Loading States**

const LegalPageSkeleton = () => (

<Layout>

<div className="max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 py-12">

<div className="mb-8 text-center">

<div className="h-8 bg-slate-700 rounded-lg w-64 mx-auto mb-4 animate-pulse" />

<div className="h-12 w-12 bg-slate-700 rounded-full mx-auto animate-pulse" />

</div>

<div className="space-y-6">

{[...Array(5)].map((\_, i) => (

<div key={i} className="space-y-3">

<div className="h-6 bg-slate-700 rounded w-48 animate-pulse" />

<div className="h-4 bg-slate-800 rounded w-full animate-pulse" />

<div className="h-4 bg-slate-800 rounded w-3/4 animate-pulse" />

</div>

))}

</div>

</div>

</Layout>

);

**5. Database Schema**

**Legal Compliance Tables**

-- User consent tracking

CREATE TABLE user\_legal\_consents (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

user\_id UUID REFERENCES users(id) NULL, -- Null for anonymous users

session\_id VARCHAR(255) NOT NULL,

consent\_type VARCHAR(50) NOT NULL, -- 'cookie', 'gdpr', 'risk'

accepted BOOLEAN NOT NULL,

ip\_address INET,

user\_agent TEXT,

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

updated\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

UNIQUE(session\_id, consent\_type)

);

-- Legal page analytics

CREATE TABLE legal\_page\_views (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

page\_name VARCHAR(100) NOT NULL,

language VARCHAR(5) NOT NULL,

user\_id UUID REFERENCES users(id) NULL,

session\_id VARCHAR(255) NOT NULL,

referrer TEXT,

time\_spent INTEGER, -- seconds

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW()

);

-- Legal content versions (for CMS if needed)

CREATE TABLE legal\_content\_versions (

id UUID PRIMARY KEY DEFAULT gen\_random\_uuid(),

page\_name VARCHAR(100) NOT NULL,

language VARCHAR(5) NOT NULL,

content JSONB NOT NULL,

version VARCHAR(20) NOT NULL,

published\_at TIMESTAMP WITH TIME ZONE,

created\_by UUID REFERENCES users(id),

created\_at TIMESTAMP WITH TIME ZONE DEFAULT NOW(),

UNIQUE(page\_name, language, version)

);

-- Indexes for performance

CREATE INDEX idx\_user\_consents\_session ON user\_legal\_consents(session\_id);

CREATE INDEX idx\_user\_consents\_type ON user\_legal\_consents(consent\_type);

CREATE INDEX idx\_legal\_views\_page ON legal\_page\_views(page\_name);

CREATE INDEX idx\_legal\_views\_session ON legal\_page\_views(session\_id);

**6. User Experience**

**Accessibility Implementation**

// Enhanced accessibility for legal pages

const AccessibleLegalPage = () => {

const { language } = useLanguage();

return (

<Layout>

<div

className={`max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 py-12 ${language === 'ar' ? 'rtl' : 'ltr'}`}

role="main"

aria-labelledby="legal-page-title"

>

<div className="mb-8 text-center">

<h1

id="legal-page-title"

className="text-3xl font-bold text-white mb-4"

tabIndex={0}

>

{title}

</h1>

<div className="flex justify-center" aria-hidden="true">

<Icon className={`h-12 w-12 ${iconColor}`} />

</div>

</div>

<nav aria-label="Page sections" className="mb-6">

<ol className="list-decimal list-inside space-y-1 text-slate-400">

{sections.map((section, index) => (

<li key={section.id}>

<a

href={`#${section.id}`}

className="hover:text-white transition-colors focus:outline-none focus:text-white"

onClick={handleSmoothScroll}

>

{section.title}

</a>

</li>

))}

</ol>

</nav>

<div className="space-y-6 text-slate-300">

{sections.map((section) => (

<section

key={section.id}

id={section.id}

aria-labelledby={`${section.id}-heading`}

tabIndex={-1}

>

<h2

id={`${section.id}-heading`}

className="text-xl font-semibold text-white mb-4"

>

{section.title}

</h2>

<div className="prose prose-invert max-w-none">

{section.content}

</div>

</section>

))}

</div>

</div>

</Layout>

);

};

**Keyboard Navigation**

// Enhanced keyboard navigation for legal pages

const useKeyboardNavigation = () => {

useEffect(() => {

const handleKeyDown = (event: KeyboardEvent) => {

// Tab to next section

if (event.key === 'Tab' && event.ctrlKey) {

event.preventDefault();

const sections = document.querySelectorAll('section[id]');

const currentIndex = Array.from(sections).findIndex(section =>

section === document.activeElement

);

const nextIndex = (currentIndex + 1) % sections.length;

(sections[nextIndex] as HTMLElement)?.focus();

}

// Return to top with Ctrl+Home

if (event.key === 'Home' && event.ctrlKey) {

event.preventDefault();

document.getElementById('legal-page-title')?.focus();

window.scrollTo({ top: 0, behavior: 'smooth' });

}

};

document.addEventListener('keydown', handleKeyDown);

return () => document.removeEventListener('keydown', handleKeyDown);

}, []);

};

**Reading Experience Enhancements**

// Reading progress indicator for long legal documents

const ReadingProgress = () => {

const [progress, setProgress] = useState(0);

useEffect(() => {

const updateProgress = () => {

const scrolled = window.scrollY;

const maxHeight = document.documentElement.scrollHeight - window.innerHeight;

const progress = (scrolled / maxHeight) \* 100;

setProgress(Math.min(progress, 100));

};

window.addEventListener('scroll', updateProgress);

return () => window.removeEventListener('scroll', updateProgress);

}, []);

return (

<div className="fixed top-0 left-0 w-full h-1 bg-slate-900 z-50">

<div

className="h-full bg-blue-500 transition-all duration-150"

style={{ width: `${progress}%` }}

/>

</div>

);

};

**7. Integration Points**

**Footer Integration Pattern**

// Both LandingPage.tsx and Layout.tsx contain legal links

const LegalFooterSection = () => {

const handleLegalLinkClick = () => {

// Track legal page navigation

analytics.track('legal\_page\_accessed', {

source: 'footer',

timestamp: new Date().toISOString()

});

// Smooth scroll to top on navigation

setTimeout(() => {

window.scrollTo({ top: 0, behavior: 'smooth' });

}, 100);

};

return (

<div>

<h3 className="text-white font-semibold mb-4 text-sm sm:text-base">Legal</h3>

<ul className="space-y-2 text-sm text-slate-400">

<li><Link to="/privacy-policy" onClick={handleLegalLinkClick}>Privacy Policy</Link></li>

<li><Link to="/terms-of-service" onClick={handleLegalLinkClick}>Terms of Service</Link></li>

<li><Link to="/risk-disclosure" onClick={handleLegalLinkClick}>Risk Disclosure</Link></li>

<li><Link to="/shariah-compliance" onClick={handleLegalLinkClick}>Shariah Compliance</Link></li>

<li><Link to="/gdpr-compliance" onClick={handleLegalLinkClick}>GDPR Compliance</Link></li>

<li><Link to="/cookie-notice" onClick={handleLegalLinkClick}>Cookie Notice</Link></li>

</ul>

</div>

);

};

**Cross-Page Navigation**

// Legal page cross-references

const LegalCrossReferences = () => {

return (

<div className="mt-8 p-4 bg-slate-800/30 border border-slate-700 rounded-lg">

<h3 className="text-white font-medium mb-3">Related Legal Documents</h3>

<div className="grid grid-cols-1 sm:grid-cols-2 gap-2 text-sm">

<Link to="/privacy-policy" className="text-blue-400 hover:text-blue-300 transition-colors">

📋 Privacy Policy

</Link>

<Link to="/risk-disclosure" className="text-amber-400 hover:text-amber-300 transition-colors">

⚠️ Risk Disclosure

</Link>

<Link to="/gdpr-compliance" className="text-green-400 hover:text-green-300 transition-colors">

🛡️ GDPR Rights

</Link>

<Link to="/cookie-notice" className="text-orange-400 hover:text-orange-300 transition-colors">

🍪 Cookie Policy

</Link>

</div>

</div>

);

};

**8. Testing Strategy**

**Component Testing**

// Legal page component tests

describe('Legal Bundle Components', () => {

describe('PrivacyPolicy', () => {

it('renders all required sections', () => {

render(<PrivacyPolicy />);

expect(screen.getByText('Information Collection')).toBeInTheDocument();

expect(screen.getByText('Use of Information')).toBeInTheDocument();

expect(screen.getByText('Data Security')).toBeInTheDocument();

});

it('supports multilingual content', () => {

const { rerender } = render(<PrivacyPolicy />);

// Test English content

expect(screen.getByText('Privacy Policy')).toBeInTheDocument();

// Switch to German

rerender(<LanguageProvider value={{language: 'de'}}><PrivacyPolicy /></LanguageProvider>);

expect(screen.getByText('Datenschutzrichtlinie')).toBeInTheDocument();

});

it('handles RTL layout for Arabic', () => {

render(<LanguageProvider value={{language: 'ar'}}><PrivacyPolicy /></LanguageProvider>);

const container = screen.getByRole('main');

expect(container).toHaveClass('rtl');

});

});

describe('Legal Page Navigation', () => {

it('scrolls to top on footer link click', async () => {

const scrollToSpy = jest.spyOn(window, 'scrollTo').mockImplementation();

render(<Layout><PrivacyPolicy /></Layout>);

const privacyLink = screen.getByText('Privacy Policy');

fireEvent.click(privacyLink);

await waitFor(() => {

expect(scrollToSpy).toHaveBeenCalledWith({

top: 0,

behavior: 'smooth'

});

});

});

});

});

**Integration Testing**

// Legal compliance flow testing

describe('Legal Compliance Integration', () => {

it('tracks consent properly', async () => {

const mockTrackConsent = jest.fn();

render(<App />);

// Navigate to cookie notice

fireEvent.click(screen.getByText('Cookie Notice'));

// Simulate consent acceptance

fireEvent.click(screen.getByText('Accept Cookies'));

expect(mockTrackConsent).toHaveBeenCalledWith({

type: 'cookie',

accepted: true,

timestamp: expect.any(String)

});

});

});

**Mock Data Structures**

// Mock legal content for testing

export const mockLegalContent = {

privacy: {

en: {

title: 'Privacy Policy',

sections: [

{ id: 'collection', title: 'Information Collection', content: 'We collect...' },

{ id: 'usage', title: 'Use of Information', content: 'We use...' }

]

},

de: {

title: 'Datenschutzrichtlinie',

sections: [

{ id: 'collection', title: 'Informationssammlung', content: 'Wir sammeln...' }

]

}

}

};

**9. Charts & Data Visualizations**

**Legal Analytics Dashboard (Admin)**

// Consent tracking visualization for admin panel

const LegalComplianceChart = () => {

const data = [

{ name: 'Cookie Consent', accepted: 85, declined: 15 },

{ name: 'GDPR Acknowledged', accepted: 92, declined: 8 },

{ name: 'Risk Disclosure', accepted: 78, declined: 22 }

];

return (

<ResponsiveContainer width="100%" height={300}>

<BarChart data={data}>

<CartesianGrid strokeDasharray="3 3" />

<XAxis dataKey="name" />

<YAxis />

<Tooltip />

<Bar dataKey="accepted" fill="#10B981" name="Accepted" />

<Bar dataKey="declined" fill="#EF4444" name="Declined" />

</BarChart>

</ResponsiveContainer>

);

};

**Legal Page Engagement Metrics**

// Time spent on legal pages

const LegalEngagementMetrics = () => {

const engagementData = [

{ page: 'Privacy Policy', avgTime: 120, views: 450 },

{ page: 'Terms of Service', avgTime: 95, views: 380 },

{ page: 'Risk Disclosure', avgTime: 180, views: 290 }

];

return (

<ResponsiveContainer width="100%" height={250}>

<ScatterChart data={engagementData}>

<CartesianGrid strokeDasharray="3 3" />

<XAxis dataKey="views" name="Page Views" />

<YAxis dataKey="avgTime" name="Avg Time (seconds)" />

<Tooltip cursor={{ strokeDasharray: '3 3' }} />

<Scatter fill="#3B82F6" />

</ScatterChart>

</ResponsiveContainer>

);

};

**10. Visual Data Elements**

**Consent Status Indicators**

const ConsentStatusIndicator = ({ consentType, status }: {

consentType: string;

status: 'accepted' | 'declined' | 'pending'

}) => {

const statusConfig = {

accepted: { color: 'text-green-400', icon: Check, bg: 'bg-green-900/20' },

declined: { color: 'text-red-400', icon: X, bg: 'bg-red-900/20' },

pending: { color: 'text-yellow-400', icon: Clock, bg: 'bg-yellow-900/20' }

};

const config = statusConfig[status];

const Icon = config.icon;

return (

<div className={`flex items-center space-x-2 p-2 rounded-lg ${config.bg}`}>

<Icon className={`h-4 w-4 ${config.color}`} />

<span className={`text-sm ${config.color}`}>

{consentType}: {status.charAt(0).toUpperCase() + status.slice(1)}

</span>

</div>

);

};

**Legal Page Icons & Visual Hierarchy**

// Consistent icon system for legal pages

const legalPageConfig = {

privacy: { icon: Shield, color: 'text-emerald-500' },

terms: { icon: FileText, color: 'text-blue-500' },

risk: { icon: AlertTriangle, color: 'text-amber-500' },

shariah: { icon: Shield, color: 'text-green-500' },

gdpr: { icon: Shield, color: 'text-blue-500' },

cookie: { icon: Cookie, color: 'text-orange-500' }

};

// Typography hierarchy

const LegalTypography = {

pageTitle: 'text-3xl font-bold text-white mb-4',

sectionTitle: 'text-xl font-semibold text-white mb-4',

subsectionTitle: 'text-lg font-medium text-white mb-2',

body: 'text-slate-300 leading-relaxed',

caption: 'text-sm text-slate-400',

highlight: 'text-white font-medium'

};

**Progress & Status Elements**

// Legal compliance progress indicator

const ComplianceProgress = ({ completedSteps, totalSteps }: {

completedSteps: number;

totalSteps: number;

}) => {

const percentage = (completedSteps / totalSteps) \* 100;

return (

<div className="w-full bg-slate-700 rounded-full h-2">

<div

className="bg-green-500 h-2 rounded-full transition-all duration-500"

style={{ width: `${percentage}%` }}

/>

<p className="text-xs text-slate-400 mt-1">

{completedSteps}/{totalSteps} legal requirements completed

</p>

</div>

);

};

**11. Security & Validation**

**Input Validation Schemas**

// Consent tracking validation

const consentSchema = z.object({

consentType: z.enum(['cookie', 'gdpr', 'risk']),

accepted: z.boolean(),

timestamp: z.string().datetime(),

sessionId: z.string().min(1),

ipAddress: z.string().ip().optional(),

userAgent: z.string().optional()

});

// Legal page analytics validation

const legalAnalyticsSchema = z.object({

page: z.enum(['privacy', 'terms', 'risk', 'shariah', 'gdpr', 'cookie']),

language: z.enum(['en', 'de', 'ar']),

timeSpent: z.number().min(0).optional(),

referrer: z.string().url().optional()

});

**Content Security & Sanitization**

// Sanitize legal content if dynamically loaded

import DOMPurify from 'dompurify';

const sanitizeLegalContent = (content: string): string => {

return DOMPurify.sanitize(content, {

ALLOWED\_TAGS: ['p', 'h1', 'h2', 'h3', 'h4', 'ul', 'ol', 'li', 'strong', 'em', 'a'],

ALLOWED\_ATTR: ['href', 'target', 'rel'],

ALLOWED\_URI\_REGEXP: /^(?:(?:https?|mailto):|[^a-z]|[a-z+.\-]+(?:[^a-z+.\-:]|$))/i

});

};

**Rate Limiting & Protection**

// Rate limiting for legal page analytics

const LEGAL\_ANALYTICS\_RATE\_LIMIT = {

windowMs: 15 \* 60 \* 1000, // 15 minutes

max: 50, // limit each IP to 50 requests per windowMs

message: 'Too many analytics requests from this IP'

};

// CSRF protection for consent submission

const consentCSRFToken = generateCSRFToken();

**12. Environment & Configuration**

**Environment Variables**

# Legal compliance configuration

LEGAL\_CONSENT\_RETENTION\_DAYS=2555 # 7 years

LEGAL\_ANALYTICS\_ENABLED=true

LEGAL\_CONTENT\_CDN\_URL=https://cdn.kurzora.com/legal

LEGAL\_ADMIN\_EMAIL=legal@kurzora.com

# Feature flags

FEATURE\_DYNAMIC\_LEGAL\_CONTENT=false

FEATURE\_LEGAL\_ANALYTICS=true

FEATURE\_CONSENT\_BANNER=true

# Third-party integrations

COOKIE\_CONSENT\_SERVICE\_KEY=your\_service\_key

GDPR\_COMPLIANCE\_API\_KEY=your\_gdpr\_key

**Legal Content Configuration**

// Configuration for legal content management

interface LegalConfig {

supportedLanguages: string[];

defaultLanguage: string;

contentVersioning: boolean;

analyticsEnabled: boolean;

consentRetentionDays: number;

requiredConsents: string[];

}

const legalConfig: LegalConfig = {

supportedLanguages: ['en', 'de', 'ar'],

defaultLanguage: 'en',

contentVersioning: true,

analyticsEnabled: process.env.LEGAL\_ANALYTICS\_ENABLED === 'true',

consentRetentionDays: parseInt(process.env.LEGAL\_CONSENT\_RETENTION\_DAYS || '2555'),

requiredConsents: ['cookie', 'gdpr']

};

**Monitoring & Analytics**

// Legal page monitoring setup

const legalPageAnalytics = {

// Track legal page performance

pageLoad: (pageName: string, loadTime: number) => {

analytics.track('legal\_page\_load\_time', {

page: pageName,

loadTime,

timestamp: new Date().toISOString()

});

},

// Track consent decisions

consentDecision: (type: string, accepted: boolean) => {

analytics.track('legal\_consent\_decision', {

consentType: type,

accepted,

timestamp: new Date().toISOString()

});

},

// Track legal link clicks

legalLinkClick: (linkType: string, source: string) => {

analytics.track('legal\_link\_click', {

linkType,

source,

timestamp: new Date().toISOString()

});

}

};

**13. Cross-Screen Data Flow**

**Legal State Synchronization**

// Legal consent affects app-wide functionality

const useLegalComplianceSync = () => {

const { hasRequiredConsents } = useLegalStore();

const { updateUserStatus } = useAuthStore();

useEffect(() => {

if (hasRequiredConsents()) {

updateUserStatus({ legalComplianceComplete: true });

}

}, [hasRequiredConsents, updateUserStatus]);

};

**Navigation State Integration**

// Legal pages integration with main navigation

const useNavigationWithLegal = () => {

const navigate = useNavigate();

const { markLegalPageViewed } = useLegalStore();

const navigateToLegalPage = (pageName: string) => {

// Track navigation

markLegalPageViewed(pageName);

// Navigate with smooth transition

navigate(`/${pageName}`, {

state: { source: 'navigation' }

});

// Scroll to top after navigation

setTimeout(() => {

window.scrollTo({ top: 0, behavior: 'smooth' });

}, 100);

};

return { navigateToLegalPage };

};

**Compliance Status Propagation**

// Legal compliance status affects other app features

interface AppComplianceState {

canAccessPremiumFeatures: boolean;

canProcessPayments: boolean;

canSendMarketing: boolean;

canUseAnalytics: boolean;

}

const useAppComplianceState = (): AppComplianceState => {

const legal = useLegalStore();

return useMemo(() => ({

canAccessPremiumFeatures: legal.hasRequiredConsents(),

canProcessPayments: legal.gdprAccepted === true,

canSendMarketing: legal.cookieConsent === true,

canUseAnalytics: legal.cookieConsent === true

}), [legal]);

};

**🎯 IMPLEMENTATION PRIORITY**

**Phase 1: Core Implementation *(Week 1)***

1. ✅ **Existing Legal Pages** - Already implemented and working
2. 🔧 **Create LegalPageWrapper** - Reduce code duplication
3. 🔧 **Add Consent Tracking** - Basic legal store implementation
4. 🔧 **Enhance Accessibility** - ARIA labels, keyboard navigation

**Phase 2: Enhanced UX *(Week 2)***

1. 🔧 **Reading Progress Indicators** - Better UX for long documents
2. 🔧 **Cross-References** - Link related legal documents
3. 🔧 **Analytics Integration** - Track legal page engagement
4. 🔧 **Performance Optimization** - Lazy loading, memoization

**Phase 3: Advanced Features *(Week 3)***

1. 🔧 **Dynamic Content Management** - Admin can update legal content
2. 🔧 **Advanced Analytics** - Legal compliance dashboards
3. 🔧 **Multi-version Support** - Handle legal content updates
4. 🔧 **Automated Compliance** - Smart consent flows

**🚀 READY FOR CURSOR IMPLEMENTATION**

**✅ Complete Technical Specifications Provided**

* All 6 legal pages analyzed with consistent patterns
* Reusable component architecture designed
* Database schema and API contracts defined
* Security, performance, and UX considerations covered
* Testing strategy and monitoring setup included

**🎯 Key Implementation Focus:**

1. **Create LegalPageWrapper component** for consistency
2. **Implement legal consent tracking store**
3. **Add accessibility enhancements**
4. **Optimize performance with lazy loading**
5. **Set up analytics and monitoring**

**📋 Total Estimated Development Time: 2-3 weeks**

This analysis provides everything needed for immediate Cursor implementation while maintaining the high-quality, production-ready standards established in your previous UI components! 🚀