

NyayaNet Frontend Architecture

Current Analysis

Your existing structure has good foundations but needs to be expanded to support all 8 menus. The theme (justice-black/constitution-gold) is excellent and should be preserved.

Proposed Folder Structure

```
text

src/
└── app/
    ├── components/
    │   ├── ui/                      # Reusable UI components
    │   │   ├── Button.tsx
    │   │   ├── Input.tsx
    │   │   ├── TextArea.tsx
    │   │   ├── Card.tsx
    │   │   ├── Badge.tsx
    │   │   ├── Tabs.tsx
    │   │   ├── Modal.tsx
    │   │   ├── Dropdown.tsx
    │   └── index.ts
    |
    ├── layout/                     # Layout components
    │   ├── Sidebar.tsx            # (Existing)
    │   ├── Header.tsx             # (Existing)
    │   ├── Footer.tsx
    │   └── MobileNotice.tsx      # (Existing)
    |
    ├── profile/                   # Profile menu components
    │   ├── ProfileHeader.tsx
    │   ├── CertificationCard.tsx
    │   ├── ProfileStats.tsx
    │   ├── ProfileTabs.tsx
    │   └── ProfileEditModal.tsx
    |
    ├── feed/                      # Feed menu components
    │   ├── CreatePost.tsx         # (Existing)
    │   ├── PostCard.tsx          # (Existing)
    │   ├── PostFilters.tsx
    │   ├── PostActions.tsx
    │   └── PostComments.tsx
    |
    ├── discussions/              # Discussion Room components
    │   ├── DiscussionCard.tsx
```

```
    |   |   |   |   |   |   | DiscussionThread.tsx  
    |   |   |   |   |   |   | ReplyCard.tsx  
    |   |   |   |   |   |   | DiscussionCreate.tsx  
    |   |   |   |   |   |   | DiscussionFilters.tsx  
    |  
    |   |   |   |   |   | network/           # Network menu components  
    |   |   |   |   |   |   | UserCard.tsx  
    |   |   |   |   |   |   | ConnectionRequest.tsx  
    |   |   |   |   |   |   | ChatWindow.tsx  
    |   |   |   |   |   |   | MessageBubble.tsx  
    |   |   |   |   |   |   | NetworkSearch.tsx  
    |  
    |   |   |   |   |   | ai-room/          # AI Room components  
    |   |   |   |   |   |   | AIModeToggle.tsx  
    |   |   |   |   |   |   | AIChatMessage.tsx  
    |   |   |   |   |   |   | AIResponseCard.tsx  
    |   |   |   |   |   |   | AIHistoryList.tsx  
    |   |   |   |   |   |   | AISessionCard.tsx  
    |  
    |   |   |   |   |   | law-search/        # Law Search components  
    |   |   |   |   |   |   | LawCategoryGrid.tsx  
    |   |   |   |   |   |   | LawActCard.tsx  
    |   |   |   |   |   |   | LawSectionView.tsx  
    |   |   |   |   |   |   | LawBookmarkButton.tsx  
    |   |   |   |   |   |   | LawSearchBar.tsx  
    |  
    |   |   |   |   |   | workspace/         # Case Workspace components  
    |   |   |   |   |   |   | NoteEditor.tsx  
    |   |   |   |   |   |   | NoteCard.tsx  
    |   |   |   |   |   |   | WorkspaceFolder.tsx  
    |   |   |   |   |   |   | NoteReferences.tsx  
    |   |   |   |   |   |   | WorkspaceSearch.tsx  
    |  
    |   |   |   |   |   | notifications/       # Notifications components  
    |   |   |   |   |   |   | NotificationItem.tsx  
    |   |   |   |   |   |   | NotificationBadge.tsx  
    |   |   |   |   |   |   | NotificationFilters.tsx  
    |   |   |   |   |   |   | NotificationActions.tsx  
    |  
    |   |   |   |   | shared/            # Shared components  
    |   |   |   |   |   | JusticeLoader.tsx # (Existing)  
    |   |   |   |   |   | CaseCard.tsx      # (Existing)  
    |   |   |   |   |   | AIAssistant.tsx # (Existing)  
    |   |   |   |   |   | SearchBar.tsx  
    |   |   |   |   |   | TagInput.tsx  
    |   |   |   |   |   | FileUpload.tsx  
    |  
    |   |   |   | pages/             # Page components (each menu)  
    |   |   |   |   | ProfilePage.tsx  
    |   |   |   |   | FeedPage.tsx  
    |   |   |   |   | DiscussionsPage.tsx
```

```
|   |   |   |   |   |   NetworkPage.tsx  
|   |   |   |   |   |   NotificationsPage.tsx  
|   |   |   |   |   |   AIRoomPage.tsx  
|   |   |   |   |   |   LawSearchPage.tsx  
|   |   |   |   |   |   WorkspacePage.tsx  
  
|   |  
|   |   lib/          # Utilities and helpers  
|   |   |   |   api/      # API client functions  
|   |   |   |   |   auth.ts  
|   |   |   |   |   profile.ts  
|   |   |   |   |   posts.ts  
|   |   |   |   |   discussions.ts  
|   |   |   |   |   network.ts  
|   |   |   |   |   ai.ts  
|   |   |   |   |   laws.ts  
|   |   |   |   |   index.ts  
  
|   |  
|   |   hooks/       # Custom React hooks  
|   |   |   |   |   useAuth.ts  
|   |   |   |   |   useFeed.ts  
|   |   |   |   |   useWebSocket.ts  
|   |   |   |   |   useSearch.ts  
|   |   |   |   |   useNotifications.ts  
  
|   |  
|   |   utils/        # Utility functions  
|   |   |   |   |   formatters.ts    # Date, text formatting  
|   |   |   |   |   validators.ts  # Form validation  
|   |   |   |   |   constants.ts   # Constants (ENUMs, config)  
|   |   |   |   |   types.ts      # TypeScript types  
  
|   |  
|   |   store/        # State management  
|   |   |   |   authStore.ts  
|   |   |   |   uiStore.ts  
|   |   |   |   notificationStore.ts  
|   |   |   |   index.ts  
  
|   |  
|   |   styles/       # Styles  
|   |   |   |   globals.css  
|   |   |   |   fonts.css      # (Existing)  
|   |   |   |   theme.css     # (Existing)  
|   |   |   |   animations.css  
|   |   |   |   components.css  
  
|   |  
|   |   App.tsx      # (Existing - Main App)  
|   |   main.tsx      # (Existing - Entry point)  
  
|  
|   assets/         # Static assets  
|   |   images/  
|   |   icons/  
|   |   fonts/
```

```
└── public/
    └── Nyaya-logo.png
```

Detailed File Specifications

1. Layout Components

components/layout/Footer.tsx

tsx

```
/**
 * Footer - Platform footer with legal disclaimers and quick links
 * Features:
 * - Copyright and legal disclaimer
 * - Quick navigation links
 * - Platform version
 * - Constitution quote of the day
 */
```

2. Profile Menu Components

components/profile/ProfileHeader.tsx

tsx

```
/**
 * ProfileHeader - User profile header with stats and actions
 * Features:
 * - User avatar with edit button
 * - Professional details (name, role, organization)
 * - Follower/Following/Post counts
 * - Follow/Connect/Message action buttons
 * - Edit profile button
 */
```

components/profile/CertificationCard.tsx

tsx

```
/**
 * CertificationCard - Display user certifications with verification status
 * Features:
```

```
* - Certification title and issuer
* - Issue/expiry dates
* - PDF preview/thumbnail
* - Verification badge
* - Download/view actions
*/
```

components/profile/ProfileTabs.tsx

tsx

```
/**
 * ProfileTabs - Tab navigation for profile sections
 * Features:
 * - Tabs: Posts, Discussions, Bookmarks, Certifications
 * - Active tab highlighting
 * - Count badges on each tab
 * - Responsive design
*/
```

3. Feed Menu Components

components/feed/PostFilters.tsx

tsx

```
/**
 * PostFilters - Filter and sort controls for feed
 * Features:
 * - Tag filter dropdown (Consumer Law, Criminal Law, etc.)
 * - Sort options (Latest, Most Liked, Most Discussed)
 * - Post type filter (All, Articles, Questions, Announcements)
 * - Following-only toggle
*/
```

components/feed/PostActions.tsx

tsx

```
/**
 * PostActions - Interactive actions for posts
 * Features:
 * - Like with reaction types (Like, Love, Insightful)
 * - Comment button with count
 * - Save to bookmarks
 * - Share options
*/
```

```
* - Follow author toggle  
*/
```

components/feed/PostComments.tsx

tsx

```
/**  
 * PostComments - Nested comments section  
 * Features:  
 * - Add comment input  
 * - Threaded replies  
 * - Comment editing  
 * - Like counts on comments  
 * - User mentions  
 */
```

4. Discussion Room Components

components/discussions/DiscussionCard.tsx

tsx

```
/**  
 * DiscussionCard - Quora-style discussion thread card  
 * Features:  
 * - Discussion title and description  
 * - Category badge (Consumer Law)  
 * - Reply count and upvotes  
 * - Best answer indicator  
 * - Follow/Save buttons  
 * - Author info with role badge  
 */
```

components/discussions/DiscussionThread.tsx

tsx

```
/**  
 * DiscussionThread - Full discussion with replies  
 * Features:  
 * - Original question at top  
 * - Threaded replies with indentation  
 * - Upvote/downvote buttons  
 * - Mark as best answer (for author)  
 * - Reply to specific comment
```

```
* - Media attachments in replies  
*/
```

components/discussions/ReplyCard.tsx

tsx

```
/**  
 * ReplyCard - Individual reply in discussion  
 * Features:  
 * - Author avatar and credentials  
 * - Reply content with formatting  
 * - Upvote count and button  
 * - Reply actions (reply, share, report)  
 * - Edited timestamp if applicable  
 * - Best answer badge  
 */
```

5. Network Menu Components

components/network/UserCard.tsx

tsx

```
/**  
 * UserCard - Professional user card for network  
 * Features:  
 * - User avatar and basic info  
 * - Role and organization  
 * - Specialization tags  
 * - Connection status (Following/Follow/Connect)  
 * - Mutual connections count  
 * - Message button  
 */
```

components/network/ChatWindow.tsx

tsx

```
/**  
 * ChatWindow - Real-time messaging interface  
 * Features:  
 * - Message list with scroll  
 * - Message input with attachments  
 * - Typing indicators  
 * - Message status (sent, delivered, read)
```

```
* - File upload preview  
* - Emoji picker  
*/
```

components/network/MessageBubble.tsx

tsx

```
/**  
 * MessageBubble - Individual chat message  
 * Features:  
 * - Message content with formatting  
 * - Timestamp  
 * - Read receipts  
 * - Message actions (copy, delete, edit)  
 * - File attachments with preview  
 * - Different styles for sent/received  
 */
```

6. AI Room Components

components/ai-room/AIModeToggle.tsx

tsx

```
/**  
 * AIModeToggle - Switch between Retrieval and Prediction modes  
 * Features:  
 * - Toggle switch between modes  
 * - Mode description tooltips  
 * - Visual indicator of active mode  
 * - Consumer Law restriction note  
 */
```

components/ai-room/AIChatMessage.tsx

tsx

```
/**  
 * AIChatMessage - Display AI query/response in chat format  
 * Features:  
 * - User query styling  
 * - AI response with citations  
 * - Confidence score display  
 * - Expandable legal references  
 * - Copy/Save actions
```

```
* - Processing time indicator
```

```
*/
```

components/ai-room/AIResponseCard.tsx

```
tsx
```

```
/**  
 * AIResponseCard - Detailed AI analysis results  
 * Features:  
 * - Predicted outcome percentage  
 * - Relevant laws with section links  
 * - Case precedents with snippets  
 * - Confidence score visualization  
 * - Save to workspace button  
 * - Export options  
 */
```

7. Law Search Components

components/law-search/LawCategoryGrid.tsx

```
tsx
```

```
/**  
 * LawCategoryGrid - Browse laws by category  
 * Features:  
 * - Grid of law categories (Consumer, Criminal, etc.)  
 * - Act count per category  
 * - Visual icon per category  
 * - Quick search by category  
 * - Bookmarked laws highlight  
 */
```

components/law-search/LawActCard.tsx

```
tsx
```

```
/**  
 * LawActCard - Display legal act information  
 * Features:  
 * - Act name and year  
 * - Section count  
 * - Category badge  
 * - Bookmark toggle  
 * - Quick view of important sections
```

```
* - Amendment status  
*/
```

components/law-search/LawSectionView.tsx

tsx

```
/**  
 * LawSectionView - Detailed view of law section  
 * Features:  
 * - Section number and title  
 * - Full text with syntax highlighting  
 * - Explanatory notes  
 * - Related sections links  
 * - Amendment history  
 * - Copy section text button  
 */
```

8. Workspace Components

components/workspace/NoteEditor.tsx

tsx

```
/**  
 * NoteEditor - Rich text editor for workspace notes  
 * Features:  
 * - Title and content editing  
 * - Basic formatting (bold, italic, lists)  
 * - Link to other entities (posts, discussions, laws)  
 * - Category and tag assignment  
 * - Auto-save functionality  
 * - Export to PDF/Word  
 */
```

components/workspace/NoteCard.tsx

tsx

```
/**  
 * NoteCard - Preview of workspace note  
 * Features:  
 * - Note title and excerpt  
 * - Last edited timestamp  
 * - Category and tags  
 * - Folder indicator
```

```
* - Archive/delete actions  
* - Reference count  
*/
```

9. Pages (Each Menu)

pages/ProfilePage.tsx

tsx

```
/**  
 * Profile Page - Complete profile interface  
 * Structure:  
 * 1. ProfileHeader (top section)  
 * 2. ProfileTabs (navigation)  
 * 3. Tab Content:  
 *     - Posts tab: Grid of user's posts  
 *     - Discussions tab: User's discussion threads  
 *     - Bookmarks tab: Saved items by type  
 *     - Certifications tab: Certification cards  
 * 4. ProfileSearchBar (local search within profile)  
 */
```

pages/FeedPage.tsx

tsx

```
/**  
 * Feed Page - Main activity stream  
 * Structure:  
 * 1. CreatePost component (top)  
 * 2. PostFilters (sorting and filtering)  
 * 3. Infinite scroll post list  
 * 4. FeedSearchBar (global feed search)  
 * 5. Trending topics sidebar  
 */
```

pages/DiscussionsPage.tsx

tsx

```
/**  
 * Discussions Page - Quora-style Q&A  
 * Structure:  
 * 1. DiscussionCreate (start new discussion)  
 * 2. DiscussionFilters (category, status, sort)
```

```
* 3. Discussion list with cards  
* 4. DiscussionSearchBar  
* 5. Popular topics sidebar  
*/
```

pages/AIRoomPage.tsx

tsx

```
/**  
 * AI Room Page - Legal AI assistance  
 * Structure:  
 * 1. AIModeToggle (Retrieval/Prediction)  
 * 2. Session list sidebar  
 * 3. Main chat interface  
 * 4. AIResponseCard for detailed results  
 * 5. AIearchBar (search past queries)  
 */
```

API Integration Layer

lib/api/profile.ts

typescript

```
/**  
 * Profile API functions  
 * Features:  
 * - getProfile(userId): Get user profile data  
 * - updateProfile(data): Update profile  
 * - uploadProfilePhoto(file): Upload profile picture  
 * - getCertifications(userId): Get user certifications  
 * - addCertification(data): Add new certification  
 * - getProfilePosts(userId, filters): Get user's posts  
 * - getProfileDiscussions(userId, filters): Get user's discussions  
 */
```

lib/api/posts.ts

typescript

```
/**  
 * Posts API functions
```

```
* Features:  
* - createPost(data): Create new post  
* - getFeed(filters, pagination): Get feed posts  
* - likePost(postId): Like/unlike post  
* - savePost(postId, folder): Save to bookmarks  
* - addComment(postId, content): Add comment  
* - searchPosts(query, filters): Search posts  
*/
```

lib/api/discussions.ts

typescript

```
/**  
 * Discussions API functions  
 * Features:  
 * - createDiscussion(data): Create new discussion  
 * - getDiscussions(filters, pagination): Get discussions  
 * - addReply(discussionId, content): Add reply  
 * - upvoteReply(replyId): Upvote reply  
 * - markBestAnswer(discussionId, replyId): Mark best answer  
 * - followDiscussion(discussionId): Follow discussion  
 */
```

lib/api/ai.ts

typescript

```
/**  
 * AI API functions  
 * Features:  
 * - createAISSession(mode): Create new AI session  
 * - sendAIQuery(sessionId, query, mode): Send query  
 * - getAIRRequestStatus(requestId): Check status  
 * - getAISessions(userId): Get user sessions  
 * - getSessionMessages(sessionId): Get chat history  
 * - searchAIQueries(query): Search past queries  
 */
```

Custom Hooks

lib/hooks/useAuth.ts

typescript

```
/**  
 * Authentication hook  
 * Features:  
 * - Current user state  
 * - Login/logout functions  
 * - Token management  
 * - Protected route check  
 * - Auto-refresh token  
 */
```

lib/hooks/useWebSocket.ts

typescript

```
/**  
 * WebSocket hook for real-time features  
 * Features:  
 * - Connect/disconnect to WS  
 * - Subscribe to events (messages, notifications)  
 * - Send messages via WS  
 * - Reconnection logic  
 * - Event handlers for real-time updates  
 */
```

lib/hooks/useNotifications.ts

typescript

```
/**  
 * Notifications hook  
 * Features:  
 * - Unread count state  
 * - Mark as read functions  
 * - Real-time notification updates  
 * - Filtering and sorting  
 * - Notification actions handler  
 */
```

State Management (Zustand)

lib/store/authStore.ts

typescript

```
/**  
 * Authentication store  
 * State:  
 * - user: Current user object  
 * - token: JWT token  
 * - isAuthenticated: Boolean  
 * - isLoading: Loading state  
 *  
 * Actions:  
 * - login(credentials)  
 * - logout()  
 * - updateUser(updates)  
 * - refreshToken()  
 */
```

lib/store/uiStore.ts

typescript

```
/**  
 * UI state store  
 * State:  
 * - sidebarOpen: Boolean  
 * - currentView: Active menu  
 * - theme: 'light' | 'dark'  
 * - notificationsOpen: Boolean  
 * - searchQuery: String  
 *  
 * Actions:  
 * - toggleSidebar()  
 * - setCurrentView(view)  
 * - openNotifications()  
 */
```

TypeScript Types

lib/utils/types.ts

typescript

```
/**  
 * Core TypeScript interfaces  
 * Features:  
 * - User types with all professional fields  
 * - Post, Discussion, Comment types  
 * - AI session and message types  
 * - Law act and section types  
 * - Notification types  
 * - API response types  
 */
```

```
export interface User {  
    id: string;  
    email: string;  
    fullName: string;  
    role: UserRole; // LAW_STUDENT, LAWYER, JUDGE, etc.  
    designation?: string;  
    organization?: string;  
    areaOfInterest: string[];  
    experienceYears: number;  
    // ... other fields  
}
```

```
export interface Post {  
    id: string;  
    userId: string;  
    title?: string;  
    content: string;  
    postType: PostType; // POST, QUESTION, ARTICLE, ANNOUNCEMENT  
    tags: string[];  
    likeCount: number;  
    commentCount: number;  
    // ... other fields  
}
```

```
export interface Discussion {  
    id: string;  
    userId: string;  
    title: string;  
    description: string;  
    discussionType: DiscussionType; // GENERAL, CASE_ANALYSIS, etc.  
    category: LawCategory; // CONSUMER_LAW, etc.  
    isResolved: boolean;  
    bestAnswerId?: string;  
    // ... other fields  
}
```

```
export interface AISession {  
    id: string;  
    userId: string;
```

```
sessionName?: string;
aiMode: AIMode; // RETRIEVAL, PREDICTION
category: LawCategory; // Always CONSUMER_LAW
isActive: boolean;
// ... other fields
}
```

Theme Preservation Guidelines

Your existing theme is excellent. To maintain consistency:

1. Color Usage:

- bg-justice-black for dark backgrounds
- text-judge-ivory for light text
- border-constitution-gold for borders
- bg-constitution-gold for primary buttons
- text-constitution-gold for gold text

2. Typography:

- font-heading for all headings (Playfair Display)
- font-body for body text (Cormorant Garamond)
- font-mono for legal citations (Courier Prime)

3. Components Patterns:

- Use aged-paper class for content cards
- Use parchment-bg for input backgrounds
- Use constitution-texture for text blocks
- Add gold border accents with border-constitution-gold/20

4. Animation Patterns:

- Use existing animation classes
- Keep loading animations justice-themed
- Smooth transitions for all interactions