BrowserFence Dashboard - Software Requirements Specification

1. Project Overview

1.1 Project Description

BrowserFence is a comprehensive Remote Browser Isolation (RBI) security dashboard that provides organizations with centralized management and monitoring of browser-based security threats. The application serves as a command center for security teams to manage isolated browser sessions, configure security policies, monitor user activity, and analyze security threats in real-time.

1.2 Purpose and Goals

- **Primary Goal**: Provide a unified interface for managing remote browser isolation infrastructure
- **Security Focus**: Enable proactive threat detection and policy enforcement for web-based activities
- **User Experience**: Deliver an intuitive, responsive dashboard for security administrators and analysts
- **Scalability**: Support enterprise-level deployments with multiple workspaces and extensive user management

1.3 Target Users

- **Security Administrators**: Configure policies, manage users, and oversee system security
- **Security Analysts**: Monitor threats, analyze logs, and investigate security incidents
- **IT Administrators**: Manage applications, integrations, and system configurations
- **End Users**: Access secured browser sessions through the RBI infrastructure

1.4 Key Value Propositions

- Centralized security policy management
- Real-time threat monitoring and analytics
- Comprehensive user and session management
- Enterprise-grade scalability and integration capabilities
- Intuitive interface reducing complexity of security operations

2. Core Features

2.1 Dashboard Overview

Purpose: Provide at-a-glance system status and security metrics
Key User Flows:

1. User logs in → Views dashboard → Reviews security metrics → Takes action on alerts
2. User navigates to specific sections via dashboard cards
3. User monitors real-time system performance and threat status

Components:

- Security metrics cards (Active Sessions, Blocked Threats, Policy Violations, User Activity)
- System status indicators
- Recent activity feed
- Quick action buttons
- Time-frame selection dropdown with mock data integration

Acceptance Criteria:

- Dashboard loads within 2 seconds
- All metrics display current data
- Cards are clickable and navigate to relevant sections
- Responsive design works on all viewport sizes
- Time frame selector updates displayed data ranges

2.2 Application Management

Purpose: Configure and manage applications available for remote browser isolation
Key User Flows:

1. Admin views application list → Creates new application → Configures settings → Saves
2. Admin toggles application status (enabled/disabled)
3. Admin edits existing application configurations

Components:

- Application list with status indicators
- Application creation panel with comprehensive form
- Enable/disable toggle switches
- Search and filtering capabilities

Acceptance Criteria:

- Applications can be created, edited, and deleted

- Status changes reflect immediately in the interface
- Form validation prevents invalid configurations
- Bulk operations supported for multiple applications

2.3 Policy Management

Purpose: Define and enforce security policies for browser isolation sessions
Key User Flows:

1. Security admin creates policy → Configures rules → Assigns to users/groups → Activates
2. Admin reviews existing policies → Modifies settings → Updates assignments
3. Admin monitors policy compliance and violations

Components:

- Policy list with status and assignment information
- Policy creation panel with rule configuration
- Policy assignment interface
- Compliance monitoring dashboard

Acceptance Criteria:

- Policies can be created with complex rule sets
- Real-time policy enforcement
- Policy conflicts are detected and resolved
- Audit trail for all policy changes

2.4 User Management

Purpose: Manage user accounts, permissions, and access controls
Key User Flows:

1. Admin adds new user → Sets permissions → Assigns groups → Activates account
2. Admin reviews user activity → Adjusts permissions → Monitors compliance
3. Admin manages user sessions and access history

Components:

- User list with detailed information table
- Add user modal with accordion-based form sections:
  - Personal Information
  - Security Settings
  - Group Assignments
  - Permissions & Access

    - Compliance & Monitoring
- User activity monitoring
- Permission management interface

Acceptance Criteria:

- Complete user lifecycle management (create, update, disable, delete)
- Role-based access control (RBAC) implementation
- User activity tracking and reporting
- Bulk user operations supported

2.5 Security Tools

2.5.1 Threat Logs

Purpose: Monitor and analyze security threats and incidents
Components: Real-time threat log viewer, filtering and search capabilities, threat categorization
Acceptance Criteria: Real-time log updates, comprehensive filtering, exportable reports

2.5.2 Templates

Purpose: Create reusable configuration templates
Components: Template library, template creation wizard, template application interface
Acceptance Criteria: Template versioning, sharing capabilities, validation before application

2.5.3 Recordings

Purpose: Manage session recordings for compliance and analysis
Components: Recording list, playback interface, retention policy management
Acceptance Criteria: Secure storage, compliance with data retention policies, search capabilities

2.5.4 App Catalog

Purpose: Browse and deploy pre-configured applications
Components: Application gallery, deployment interface, configuration options
Acceptance Criteria: Easy browsing and deployment, configuration validation, rollback capabilities

2.6 Insights and Analytics

Purpose: Provide comprehensive security analytics and reporting
Key User Flows:

1. Analyst views insights → Selects metrics → Generates reports → Exports data

2. Analyst creates custom dashboards → Configures alerts → Monitors trends

Components:

- Interactive charts and graphs
- Geographic threat mapping
- Custom report builder
- Alert configuration interface

Acceptance Criteria:

- Real-time data visualization
- Customizable reporting periods
- Export capabilities (PDF, CSV, JSON)
- Automated report scheduling

3. UI/UX Design System

3.1 Design System Foundation

Framework: Preline UI design system with TailwindCSS
Typography: Inter font family with defined weight and size scales
Color Palette:

- Primary: Blue-based theme (#030213, `#2563eb`, `#1d4ed8`)
- Secondary: Gray scale for neutral elements
- Status Colors: Green (success), Red (danger), Yellow (warning), Blue (info)

3.2 Layout Structure

3.2.1 Top Navigation Bar

Components:

- Logo and application branding (BrowserFence with Shield icon)
- Workspace selector with badges
- Navigation dropdowns (Workspaces, Projects, Integrations)
- Search functionality with animation
- Notification dropdown
- Settings dropdown
- User profile dropdown

Responsive Behavior:

- Desktop: Full navigation visible

- Tablet: Condensed navigation with dropdowns
- Mobile: Hamburger menu with sidebar overlay

3.2.2 Sidebar Navigation

States:

- Expanded (desktop default): Full labels and icons visible
- Collapsed (tablet): Icons only with tooltips
- Hidden (mobile): Off-canvas with overlay

Sections:

- Overview: Dashboard, Applications, Policies, Insights, User Management
- Security Tools: Threat Logs, Templates, Recordings, App Catalog

Interaction:

- Smooth transitions between states
- Active section highlighting
- Hover effects and focus states

3.2.3 Main Content Area

Layout: Responsive grid system adapting to content type
Spacing: Consistent padding and margins using design tokens
Components: Cards, tables, forms, charts, and modal panels

3.3 Component Specifications

3.3.1 Interactive Elements

Buttons:

- Primary (blue background), Secondary (outline), Ghost (transparent)
- Size variants: sm, md, lg
- States: default, hover, active, disabled, loading

Form Elements:

- Input fields with validation states
- Dropdowns with search capabilities
- Toggle switches for enable/disable actions
- Multi-select components for assignments

Data Display:

- Tables with sorting, filtering, and pagination
- Cards with metrics and status indicators
- Progress indicators for compliance tracking
- Badge components for status and categorization

3.3.2 Modal and Panel System

Creation Panels:

- Full-viewport sliding panels for complex forms
- Accordion-based organization for logical groupings
- Form validation with real-time feedback
- Save/cancel actions with confirmation

Dialog Modals:

- Confirmation dialogs for destructive actions
- Information displays for detailed views
- Alert notifications for system messages

3.4 Responsive Design Requirements

Breakpoints:

- Mobile: < 768px
- Tablet: 768px - 1024px
- Desktop: > 1024px

Adaptive Behavior:

- Mobile-first approach with progressive enhancement
- Touch-friendly interface elements
- Optimized navigation for smaller screens
- Content reflow for different viewport sizes

4. System Architecture

4.1 Frontend Architecture

Framework: React 18 with TypeScript
State Management: Custom hooks for application and policy management
Styling: TailwindCSS v4 with custom design tokens
Component Library: ShadCN/UI components with Preline customizations

4.2 Component Structure

```
App.tsx (Main application shell)
├── Navigation Components
│   ├── TopNavigation (header with dropdowns)
│   ├── Sidebar (collapsible navigation)
│   └── Breadcrumb (page context)
├── Content Components
│   ├── DashboardContent
│   ├── ApplicationsContent
│   ├── PoliciesContent
│   ├── UserManagementContent
│   └── Security Tool Components
├── Creation Panels
│   ├── ApplicationCreationPanel
│   ├── PolicyCreationPanel
│   └── Template/User Creation Components
└── Shared Components
    ├── UI Components (buttons, inputs, tables)
    ├── Chart Components (analytics visualization)
    └── Utility Components (loading, error states)
```

4.3 State Management Patterns

Custom Hooks:

- `useApplicationManagement`: Handles application CRUD operations and state
- `usePolicyManagement`: Manages policy configurations and assignments
- Sidebar and responsive behavior managed in main App component

Data Flow:

- Unidirectional data flow from parent to child components
- Event handling passed down as props
- Local state for UI interactions, global state for data management

4.4 Integration Requirements

Backend API: RESTful API for data operations (mocked in current implementation)
Real-time Updates: WebSocket connections for live monitoring
External Integrations: Support for SIEM, LDAP, and other security tools

5. Data Models

5.1 Core Entities

5.1.1 User Entity

```
interface User {
  id: string;
  email: string;
  firstName: string;
  lastName: string;
  role: 'admin' | 'analyst' | 'user';
  department: string;
  status: 'active' | 'inactive' | 'suspended';
  lastLogin: Date;
  createdAt: Date;
  permissions: Permission[];
  groupMemberships: GroupMembership[];
  complianceScore: number;
  riskLevel: 'low' | 'medium' | 'high';
}
```

5.1.2 Application Entity

```
interface Application {
  id: string;
  name: string;
  description: string;
  url: string;
  category: string;
  status: 'enabled' | 'disabled';
  isolationPolicy: Policy;
  allowedUsers: User[];
  configuration: ApplicationConfig;
  createdAt: Date;
  updatedAt: Date;
}
```

5.1.3 Policy Entity

```
interface Policy {
  id: string;
  name: string;
  description: string;
  type: 'security' | 'access' | 'compliance';
  rules: PolicyRule[];
  assignedUsers: User[];
  assignedGroups: Group[];
  status: 'active' | 'inactive' | 'draft';
  priority: number;
  createdAt: Date;
```

```
  updatedAt: Date;
}
```

5.1.4 Session Entity

```
interface BrowserSession {
  id: string;
  userId: string;
  applicationId: string;
  startTime: Date;
  endTime?: Date;
  duration: number;
  status: 'active' | 'completed' | 'terminated';
  ipAddress: string;
  userAgent: string;
  activities: SessionActivity[];
  threats: ThreatEvent[];
}
```

5.2 Workspace and Organizational Models

5.2.1 Workspace Entity

```
interface Workspace {
  id: string;
  name: string;
  description: string;
  type: 'Development' | 'Production' | 'Testing';
  icon: React.ComponentType;
  settings: WorkspaceSettings;
  members: WorkspaceMember[];
  projects: Project[];
  createdAt: Date;
}
```

5.2.2 Project Entity

```
interface RBIProject {
  id: string;
  name: string;
  description: string;
  workspaceId: string;
  status: 'Active' | 'In Progress' | 'Completed' | 'On Hold';
  priority: 'High' | 'Medium' | 'Low';
  progress: number;
  icon: React.ComponentType;
```

```
  startDate: Date;
  targetDate: Date;
  assignedUsers: User[];
}
```

5.3 Security and Monitoring Models

5.3.1 Threat Event Entity

```
interface ThreatEvent {
  id: string;
  sessionId: string;
  type: 'malware' | 'phishing' | 'data_exfiltration' | 'policy_violation';
  severity: 'low' | 'medium' | 'high' | 'critical';
  description: string;
  detectionTime: Date;
  source: string;
  status: 'detected' | 'investigating' | 'resolved' | 'false_positive';
  responseActions: ResponseAction[];
}
```

5.3.2 Integration Entity

```
interface Integration {
  id: string;
  name: string;
  type: string;
  category: 'SIEM' | 'Identity' | 'Productivity' | 'Developer Tools';
  status: 'connected' | 'disconnected' | 'error';
  icon: React.ComponentType;
  lastSync: string;
  configuration: IntegrationConfig;
  metrics: IntegrationMetrics;
}
```

5.4 Data Relationships

- **Users** belong to **Workspaces** and can be assigned to **Projects**
- **Applications** are governed by **Policies** and generate **Sessions**
- **Sessions** can contain multiple **Threat Events** and **Activities**
- **Policies** can be assigned to **Users**, **Groups**, or **Applications**
- **Integrations** sync data across **Workspaces** and external systems

6. User Roles & Permissions

6.1 Role Definitions

6.1.1 Super Administrator

Capabilities:

- Full system access and configuration
- User and workspace management
- System-wide policy creation and enforcement
- Integration and API management
- Audit and compliance reporting
- System maintenance and updates

6.1.2 Security Administrator

Capabilities:

- Policy creation and management
- User security settings and permissions
- Threat monitoring and response
- Security tool configuration
- Compliance reporting
- Application security configuration

6.1.3 Security Analyst

Capabilities:

- Threat log monitoring and analysis
- Security incident investigation
- Report generation and analysis
- User activity monitoring
- Policy compliance review
- Limited application access management

6.1.4 IT Administrator

Capabilities:

- Application management and deployment
- User account creation and basic management
- Template management
- Integration monitoring
- Basic reporting and analytics

- System performance monitoring

6.1.5 End User

Capabilities:

- Access assigned applications through RBI
- View personal session history
- Basic profile management
- Access to help and support resources

6.2 Permission Matrix

Feature	Super Admin	Sec Admin	Sec Analyst	IT Admin	End User
Dashboard Access	Full	Full	Limited	Limited	Personal
User Management	Full	Limited	View Only	Basic	None
Policy Management	Full	Full	View Only	None	None
Application Management	Full	Security Config	View Only	Full	Assigned Only
Threat Monitoring	Full	Full	Full	View Only	None
System Settings	Full	Limited	None	Limited	None
Integrations	Full	Limited	View Only	Monitor	None
Reporting	Full	Full	Limited	Limited	Personal

6.3 Access Control Implementation

- **Role-Based Access Control (RBAC)**: Primary authorization model
- **Attribute-Based Access Control (ABAC)**: For fine-grained permissions
- **Multi-Factor Authentication (MFA)**: Required for administrative roles
- **Session Management**: Automatic timeout and re-authentication
- **Audit Logging**: All permission changes and access attempts logged

7. Design System Specifications

7.1 Typography System (Preline Standard)

Primary Font: Inter
Font Weights: 300 (light) to 900 (black)
Size Scale:

- xs: 12px (0.75rem)
- sm: 14px (0.875rem) - Base application text
- base: 16px (1rem)
- lg: 18px (1.125rem)
- xl: 20px (1.25rem)
- 2xl: 24px (1.5rem)
- 3xl: 30px (1.875rem)
- 4xl: 36px (2.25rem)

7.2 Color Palette

Primary Colors:

- Blue: `#2563eb` (primary), `#1d4ed8` (hover), `#1e40af` (active)
- Background: `#ffffff` (light), `#f8fafc` (secondary)
- Text: #111827 (primary), `#6b7280` (secondary), `#9ca3af` (muted)

Status Colors:

- Success: `#10b981` (green)
- Warning: `#f59e0b` (amber)
- Error: `#ef4444` (red)
- Info: `#3b82f6` (blue)

Component Colors:

- Border: rgba(0, 0, 0, 0.1)
- Input Background: `#f3f3f5`
- Card Background: `#ffffff`
- Sidebar: `#f8fafc`

7.3 Spacing System

Base Unit: 4px
Scale: 1 (4px), 2 (8px), 3 (12px), 4 (16px), 6 (24px), 8 (32px), 12 (48px), 16 (64px)
Component Spacing:

- Card padding: 24px (6)
- Button padding: 12px 16px (3 4)
- Form field spacing: 16px (4)
- Section margins: 32px (8)

7.4 Component Specifications

7.4.1 Buttons

```
/* Primary Button */
.btn-primary {
  background: #2563eb;
  color: #ffffff;
  border-radius: 6px;
  padding: 8px 16px;
  font-weight: 500;
  font-size: 14px;
}

/* Secondary Button */
.btn-secondary {
  background: transparent;
  color: #2563eb;
  border: 1px solid #e5e7eb;
  border-radius: 6px;
  padding: 8px 16px;
}
```

7.4.2 Form Elements

- Input height: 40px (10)
- Border radius: 6px
- Border color: `#e5e7eb`
- Focus ring: 2px `#3b82f6` with 0.2 opacity
- Label font-weight: 500

7.4.3 Cards and Containers

- Border radius: 8px
- Border: 1px solid `#e5e7eb`
- Background: `#ffffff`
- Padding: 24px
- Shadow: 0 1px 3px rgba(0, 0, 0, 0.1)

7.5 Icon System

Icon Library: Lucide React
Standard Size: 16px (4) for inline, 20px (5) for standalone
Color: Inherits from parent or uses semantic colors
Usage: Consistent placement and sizing across components

8. Interaction & Behavior Specifications

8.1 Navigation Behavior

8.1.1 Sidebar Navigation

Responsive States:

- Desktop Expanded (>1024px): Full sidebar with labels
- Desktop Collapsed (>1024px): Icon-only sidebar with tooltips
- Mobile Hidden (<768px): Off-canvas sidebar with overlay

Transitions:

- Sidebar width changes: 300ms ease-in-out
- Icon to label transitions: 200ms ease
- Mobile slide-in: 250ms cubic-bezier(0.4, 0, 0.2, 1)

User Interactions:

- Toggle button changes sidebar state
- Active section highlighted with blue background
- Hover effects on navigation items
- Auto-close sidebar on mobile after navigation

8.1.2 Top Navigation

Search Functionality:

- Animated expansion from 40px to 280px
- Smooth 300ms transition with easing
- Auto-focus on input when expanded
- Escape key closes search
- Click outside closes search

Dropdown Menus:

- Slide-down animation with fade-in
- Click outside to close
- Keyboard navigation support
- Proper focus management

8.2 Data Interaction Patterns

8.2.1 Table Interactions

Sorting: Click column headers to sort, visual indicators for sort direction
Filtering: Real-time filtering as user types
Pagination: Configurable page sizes with navigation controls
Row Selection: Individual and bulk selection with visual feedback
Inline Editing: Click-to-edit for appropriate fields

8.2.2 Form Interactions

Validation: Real-time validation with error messages
Auto-save: Draft saving for long forms
Field Dependencies: Dynamic field visibility based on selections
Progress Indication: Multi-step forms show progress

8.3 Modal and Panel Behavior

8.3.1 Creation Panels

Opening: Slide in from right side of screen
Sizing: Full viewport (96vw × 96vh)
Content Organization: Accordion sections for logical grouping
Closing:

- X button in top-right
- Cancel button dismisses with confirmation
- Click outside does not close (prevents accidental loss)

Form Behavior:

- Real-time validation
- Section completion indicators
- Auto-save functionality
- Confirmation on unsaved changes

8.3.2 Confirmation Dialogs

Destructive Actions: Always require confirmation
Information Display: Non-blocking informational modals
Loading States: Show progress for long-running operations

8.4 Animation and Transitions

Principles:

- Smooth, purposeful animations
- Consistent timing (200-300ms for most transitions)
- Easing functions: ease-in-out for most, cubic-bezier for complex

Specific Animations:

- Page transitions: Smooth scroll to top
- Card hover effects: Subtle lift and shadow
- Button interactions: Scale and color transitions
- Loading states: Spinner or skeleton loading
- Toast notifications: Slide-in from top-right

8.5 Error Handling and Feedback

8.5.1 Error States

Form Errors: Inline validation with red styling and clear messages
Network Errors: Toast notifications with retry options
System Errors: Full-page error states with recovery options
Permission Errors: Informative messages with next steps

8.5.2 Success Feedback

Action Confirmation: Toast notifications for successful operations
Visual Feedback: Green checkmarks and success styling
Progress Indicators: Show completion status for multi-step processes

8.5.3 Loading States

Page Loading: Skeleton screens while content loads
Action Loading: Button spinners for form submissions
Data Loading: Table and card loading states
Progressive Loading: Load critical content first

9. Non-Functional Requirements

9.1 Performance Requirements

9.1.1 Load Times

- Initial page load: < 3 seconds on 3G connection
- Navigation between sections: < 1 second
- Dashboard data refresh: < 2 seconds
- Search results: < 500ms
- Form submissions: < 2 seconds with feedback within 200ms

9.1.2 Scalability

- Support 10,000+ concurrent users

- Handle datasets with 100,000+ records
- Support workspaces with 50,000+ users
- Maintain performance with 1M+ threat events

9.1.3 Resource Usage

- Memory usage: < 100MB for typical session
- Network efficiency: Minimize API calls through caching
- Bundle size: < 1MB for initial load, code splitting for routes
- CPU usage: Efficient rendering for large datasets

9.2 Security Requirements

9.2.1 Authentication and Authorization

- Multi-factor authentication for admin roles
- JWT tokens with secure storage
- Session timeout after 8 hours of inactivity
- Role-based access control (RBAC)
- Attribute-based access control (ABAC) for fine-grained permissions

9.2.2 Data Security

- Encryption in transit (HTTPS/TLS 1.3)
- Encryption at rest for sensitive data
- Input validation and sanitization
- XSS and CSRF protection
- Secure API key management

9.2.3 Privacy and Compliance

- GDPR compliance for user data
- SOC 2 Type II compliance
- Data retention policies
- Audit logging for all administrative actions
- Right to data deletion

9.3 Accessibility Requirements

9.3.1 WCAG 2.1 AA Compliance

- Keyboard navigation for all interactive elements
- Screen reader compatibility

- Color contrast ratios meeting AA standards
- Alternative text for images and icons
- Focus indicators visible and appropriate

9.3.2 Assistive Technology Support

- ARIA labels and roles properly implemented
- Semantic HTML structure
- Skip navigation links
- Proper heading hierarchy
- Form labels and error announcements

9.4 Browser and Device Support

9.4.1 Browser Support

- Chrome 90+ (primary)
- Firefox 88+ (secondary)
- Safari 14+ (secondary)
- Edge 90+ (secondary)

9.4.2 Device Support

- Desktop: Full functionality
- Tablet: Optimized responsive experience
- Mobile: Core functionality with touch optimization
- Screen sizes: 320px to 3840px width

9.5 Reliability and Availability

9.5.1 Uptime Requirements

- 99.9% availability during business hours
- Planned maintenance windows: 4 hours/month maximum
- Recovery time objective (RTO): 4 hours
- Recovery point objective (RPO): 1 hour

9.5.2 Error Recovery

- Graceful degradation for API failures
- Offline capability for viewing cached data
- Automatic retry mechanisms for failed operations
- User-friendly error messages with recovery guidance

10. Technical Implementation Details

10.1 Frontend Technology Stack

```
{
  "framework": "React 18 with TypeScript",
  "styling": "TailwindCSS v4 with custom design tokens",
  "components": "ShadCN/UI with Preline customizations",
  "animations": "Motion/React (formerly Framer Motion)",
  "icons": "Lucide React",
  "charts": "Recharts for data visualization",
  "state": "Custom hooks with React state",
  "routing": "React Router (if multi-page)",
  "forms": "React Hook Form with validation"
}
```

10.2 Code Organization

```
src/
├── components/           # Reusable UI components
│   ├── ui/              # ShadCN base components
│   ├── navigation/      # Navigation-specific components
│   └── figma/           # Figma-imported components
├── pages/               # Page-level components
├── hooks/               # Custom React hooks
├── data/                # Mock data and constants
├── types/               # TypeScript type definitions
├── utils/               # Helper functions
└── styles/              # Global styles and design tokens
```

10.3 State Management Patterns

Custom Hooks Strategy:

- `useApplicationManagement` : Application CRUD operations
- `usePolicyManagement` : Policy configuration and assignment
- Individual hooks for complex forms and UI state
- Context providers for global state when needed

10.4 Performance Optimizations

- **Code Splitting**: Route-based splitting for reduced initial bundle
- **Lazy Loading**: Dynamic imports for heavy components
- **Memoization**: React.memo and useMemo for expensive calculations

- **Virtual Scrolling**: For large data lists and tables
- **Image Optimization**: Responsive images with fallback handling

10.5 Development Workflow

File Naming Conventions:

- Components: PascalCase (e.g., `DashboardContent.tsx`)
- Hooks: camelCase starting with 'use' (e.g., `useApplicationManagement.ts`)
- Utilities: camelCase (e.g., `helpers.ts`)
- Types: camelCase interfaces (e.g., `User`, `Application`)

Import Organization:

```
// External dependencies
import React, { useState } from "react";
import { motion } from "motion/react";

// UI components
import { Button } from "./components/ui/button";

// Internal components
import DashboardContent from "./components/DashboardContent";

// Hooks and utilities
import { useApplicationManagement } from "./hooks/useApplicationManagement";

// Types and data
import type { User } from "./types";
import { navigationItems } from "./data/navigationData";
```

11. API and Integration Requirements

11.1 RESTful API Specifications

11.1.1 Authentication Endpoints

```
POST /api/auth/login
POST /api/auth/logout
POST /api/auth/refresh
GET  /api/auth/profile
PUT  /api/auth/profile
POST /api/auth/change-password
```

11.1.2 User Management Endpoints

```
GET    /api/users              # List users with pagination
POST   /api/users              # Create new user
GET    /api/users/:id          # Get user details
PUT    /api/users/:id          # Update user
DELETE /api/users/:id          # Delete user
GET    /api/users/:id/sessions # User session history
PUT    /api/users/:id/status   # Enable/disable user
```

11.1.3 Application Management Endpoints

```
GET    /api/applications       # List applications
POST   /api/applications       # Create application
GET    /api/applications/:id   # Get application
PUT    /api/applications/:id   # Update application
DELETE /api/applications/:id   # Delete application
PUT    /api/applications/:id/status # Toggle application status
```

11.1.4 Policy Management Endpoints

```
GET    /api/policies           # List policies
POST   /api/policies           # Create policy
GET    /api/policies/:id       # Get policy
PUT    /api/policies/:id       # Update policy
DELETE /api/policies/:id       # Delete policy
POST   /api/policies/:id/assign # Assign policy to users/groups
```

11.2 Real-Time Communication

WebSocket Connections:

- Threat event notifications
- User activity monitoring
- System status updates
- Real-time dashboard metrics

Message Format:

```
{
  "type": "THREAT_DETECTED",
  "payload": {
    "sessionId": "uuid",
    "threatType": "malware",
    "severity": "high",
```

```
    "timestamp": "2024-01-15T10:30:00Z"
  }
}
```

11.3 External Integrations

11.3.1 SIEM Integration

- Send threat events and logs
- Receive alerts and triggers
- Bi-directional data synchronization
- Standard formats: CEF, STIX/TAXII

11.3.2 Identity Provider Integration

- LDAP/Active Directory synchronization
- SAML/OIDC single sign-on
- User attribute mapping
- Group membership synchronization

11.3.3 Productivity Tool Integration

- Microsoft 365 integration
- Google Workspace integration
- Slack notifications
- Custom webhook support

12. Future Extensions and Roadmap

12.1 Planned Features

12.1.1 Advanced Analytics

- Machine learning-based threat detection
- Predictive analytics for risk assessment
- Custom dashboard creation
- Advanced reporting and visualization

12.1.2 Mobile Application

- iOS and Android native apps
- Core functionality for mobile users
- Push notifications for critical alerts
- Offline capability for viewing reports

12.1.3 API Gateway

- Public API for third-party integrations
- Rate limiting and quota management
- API key management
- Developer portal and documentation

12.2 Scalability Considerations

12.2.1 Multi-Tenancy

- Tenant isolation and data segregation
- Customizable branding per tenant
- Tenant-specific configurations
- Usage and billing analytics

12.2.2 Global Deployment

- Multi-region support
- Content delivery network (CDN) integration
- Localization and internationalization
- Regional compliance requirements

12.3 Technology Evolution

12.3.1 Frontend Modernization

- Progressive Web App (PWA) capabilities
- Edge computing for improved performance
- Advanced caching strategies
- Component library extraction

12.3.2 Backend Enhancements

- Microservices architecture
- Event-driven architecture
- Container orchestration
- Serverless functions for specific operations

13. Implementation Guidelines

13.1 Development Phases

Phase 1: Core Infrastructure (Weeks 1-4)

- Set up development environment and build tools
- Implement basic layout and navigation
- Create foundational components and design system
- Set up routing and state management

Phase 2: Core Features (Weeks 5-12)

- Implement Dashboard with basic metrics
- Build Application and Policy management
- Create User Management interface
- Develop creation panels and forms

Phase 3: Advanced Features (Weeks 13-20)

- Add Security Tools (Threat Logs, Templates, etc.)
- Implement Insights and Analytics
- Build notification and alert systems
- Add responsive design and mobile optimization

Phase 4: Polish and Integration (Weeks 21-24)

- Performance optimization
- Security hardening
- Accessibility compliance
- Integration testing and documentation

13.2 Quality Assurance

13.2.1 Testing Strategy

- Unit tests for utility functions and hooks
- Component testing for UI components
- Integration testing for user flows
- End-to-end testing for critical paths
- Performance testing for large datasets

13.2.2 Code Quality Standards

- TypeScript strict mode
- ESLint and Prettier configuration
- Code review requirements
- Automated security scanning
- Documentation requirements

13.3 Deployment Strategy

13.3.1 Environment Setup

- Development: Local development with hot reloading
- Staging: Pre-production testing environment
- Production: High-availability deployment
- Feature branches: Isolated feature development

13.3.2 CI/CD Pipeline

- Automated building and testing
- Security vulnerability scanning
- Performance regression testing
- Automated deployment to staging
- Manual approval for production deployment

14. Success Metrics and KPIs

14.1 User Experience Metrics

- Page load times < 3 seconds
- Task completion rates > 95%
- User satisfaction scores > 4.5/5
- Support ticket reduction by 40%

14.2 Technical Performance Metrics

- System uptime > 99.9%
- API response times < 200ms
- Error rates < 0.1%
- Security vulnerability resolution within 24 hours

14.3 Business Impact Metrics

- User adoption rate > 80% within 6 months
- Productivity increase in security operations
- Reduction in security incident response time
- Cost savings from automation

Appendices

A. Glossary of Terms

- **RBI**: Remote Browser Isolation - Security technology that executes web browser sessions in isolated environments
- **SIEM**: Security Information and Event Management - System for analyzing security alerts
- **RBAC**: Role-Based Access Control - Access control method based on user roles
- **ABAC**: Attribute-Based Access Control - Fine-grained access control using attributes

B. Reference Links

- Preline UI Documentation: Component specifications and design patterns
- WCAG 2.1 Guidelines: Accessibility compliance requirements
- OWASP Top 10: Security vulnerability prevention
- React Documentation: Framework-specific best practices

C. Contact Information

- Product Owner: Responsible for feature requirements and prioritization
- Technical Lead: Oversees architecture and technical decisions
- UX Designer: Ensures design consistency and user experience
- Security Team: Reviews security requirements and implementations

---

This requirements document serves as the single source of truth for rebuilding the BrowserFence dashboard application. All features, specifications, and requirements outlined here should be implemented to ensure consistency with the original design and functionality.