# **Product Requirements Document (PRD)**

## **ChatApp: Enterprise AI Chat Application**

### **Document Information**

* **Product Name**: ChatApp
* **Version**: 1.0
* **Date**: June 30, 2025

## **1. Product Executive Summary**

### **1.1 Overview**

ChatApp is an enterprise-grade conversational AI application that seamlessly integrates Microsoft Azure Active Directory authentication with advanced AI capabilities. The product delivers a secure, scalable chat interface that enables authenticated users to interact with state-of-the-art language models in a protected corporate environment.

### **1.2 Business Objectives**

* **Primary Goal**: Provide secure AI-powered conversational capabilities for enterprise users
* **Secondary Goals**:
* Demonstrate enterprise-ready AI integration patterns
* Establish foundation for future AI-powered business applications
* Enable secure knowledge sharing and assistance within organizations

### **1.3 Value Proposition**

* **Security First**: Enterprise-grade authentication and authorization
* **Seamless Integration**: Native Microsoft ecosystem compatibility
* **Modern UI/UX**: Responsive, accessible interface design
* **Scalable Architecture**: Cloud-native, microservices-ready design

## **2. Product Vision and Strategy**

### **2.1 Vision Statement**

To create the most secure and user-friendly AI chat platform that seamlessly integrates with enterprise identity systems, enabling organizations to leverage AI capabilities while maintaining strict security and compliance standards.

### **2.2 Strategic Goals**

* Launch MVP with core chat and authentication features
* Implement advanced conversation management and history
* Add multi-model AI support and custom skills
* Enterprise deployment tools and admin console

### **2.3 Target Market**

* **Primary**: Enterprise organizations using Microsoft 365 ecosystem
* **Secondary**: Educational institutions with Azure AD implementation
* **Tertiary**: Development teams learning enterprise AI integration

## **3. Product Architecture and Technical Requirements**

### **3.1 High-Level Architecture**

┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐  
│ Frontend │ │ Backend API │ │ External │  
│ (Next.js) │◄──►│ (Next.js) │◄──►│ Services │  
│ │ │ │ │ │  
│ • React UI │ │ • Authentication│ │ • Azure AD │  
│ • MSAL Auth │ │ • API Gateway │ │ • Lab45 AI │  
│ • TypeScript │ │ • Request Proxy │ │ • Monitoring │  
└─────────────────┘ └─────────────────┘ └─────────────────┘

### **3.2 Technology Stack**

#### **Frontend**

* **Framework**: Next.js 15.1.0 with React 18
* **Language**: TypeScript 5.x
* **Styling**: Tailwind CSS 3.4.1
* **Authentication**: Azure MSAL React 2.2.0
* **Build Tool**: Next.js built-in bundler

#### **Backend**

* **Runtime**: Node.js (via Next.js API routes)
* **Language**: TypeScript
* **Authentication**: Azure MSAL Browser 3.28.0
* **API Integration**: Native Fetch API

#### **Infrastructure**

* **Hosting**: Vercel/Azure App Service
* **CDN**: Built-in Next.js optimization
* **Monitoring**: Application Insights
* **Security**: Azure Security Center

### **3.3 Data Flow Architecture**

**1.User Authentication**:

User → Azure AD → MSAL → Access Token → Frontend State

**2.Chat Interaction**:

User Input → Frontend → API Route → Lab45 AI → Streaming Response → UI Update

**3.Security Layer**:

All Requests → Token Validation → Authorization Check → API Access

## **4. Functional Requirements**

### **4.1 Core Features**

#### **4.1.1 User Authentication**

* Users must authenticate using Microsoft Azure AD credentials
* System shall support single sign-on (SSO) capabilities
* Authentication tokens shall be securely cached in session storage
* Users shall be able to sign out and clear authentication state

#### **4.1.2 Chat Interface**

* Users shall interact through a conversational chat interface
* System shall display conversation history in chronological order
* User messages shall be visually distinguished from AI responses
* System shall provide real-time typing indicators during AI processing

#### **4.1.3 AI Integration**

* System shall integrate with ChatApp AI service using GPT-3.5-turbo-16k model
* AI responses shall be streamed in real-time for better user experience
* System shall handle AI service errors gracefully with user-friendly messages
* AI model parameters shall be configurable (temperature, max tokens)

### **4.2 User Interface Requirements**

#### **4.2.1 Responsive Design**

* Interface shall be fully responsive across desktop, tablet, and mobile
* Touch-friendly controls for mobile devices
* Keyboard navigation support for accessibility

#### **4.2.2 Visual Design**

* Modern, clean interface following Microsoft design principles
* Dark/light theme support based on system preferences
* Loading states and progress indicators for all async operations

## **5. Advanced Features**

### **5.1 Phase 2 Features**

#### **5.1.1 Conversation Management**

* Conversation history persistence across sessions
* Conversation search and filtering capabilities
* Export conversation transcripts (PDF, TXT formats)
* Conversation sharing with team members

#### **5.1.2 Enhanced AI Capabilities**

* Multi-model support (GPT-4, Claude, custom models)
* Context-aware responses using conversation history
* File upload and analysis capabilities
* Custom AI skills and prompt templates

### **5.2 Phase 3 Features**

#### **5.2.1 Collaboration Features**

* Team chat rooms with shared AI assistant
* Real-time collaborative conversations
* AI response rating and feedback system
* Knowledge base integration

#### **5.2.2 Administrative Features**

* Admin dashboard for usage analytics
* User management and permission controls
* Audit logging and compliance reporting
* Custom branding and configuration

## **6. Integration Requirements**

### **6.1 Microsoft Ecosystem Integration**

* Native Azure Active Directory integration
* Microsoft Graph API integration for user profiles
* Office 365 applications integration (Teams, Outlook)
* Azure Application Insights for monitoring

### **6.2 Third-Party Services**

* ChatApp AI API integration with proper error handling
* Webhook support for external system notifications
* REST API endpoints for third-party integrations
* OAuth 2.0 support for additional identity providers

### **6.3 Enterprise Systems**

* LDAP/Active Directory synchronization
* SAML 2.0 support for federated authentication
* Enterprise logging systems integration
* Compliance and governance tool integration

## **7. API Requirements**

### **7.1 Internal API Specifications**

#### **7.1.1 Authentication Endpoints**

// Token validation  
POST /api/auth/validate  
Headers: Authorization: Bearer <token>  
Response: { valid: boolean, user: UserProfile }  
  
// Token refresh  
POST /api/auth/refresh  
Body: { refreshToken: string }  
Response: { accessToken: string, expiresIn: number }

#### **7.1.2 Chat Endpoints**

// Send message  
POST /api/completions  
Headers: Authorization: Bearer <token>  
Body: { prompt: string, context?: string }  
Response: Stream<{ data: { content: string } }>  
  
// Get conversation history  
GET /api/conversations/{id}  
Headers: Authorization: Bearer <token>  
Response: { messages: Message[], metadata: ConversationMeta }

### **7.2 External API Integration**

#### **7.2.1 ChatApp AI Service**

* **Endpoint**: <https://api.lab45.ai/v1.1/skills/completion/query>
* **Authentication**: Bearer token passthrough
* **Rate Limiting**: 100 requests/minute per user
* **Timeout**: 30 seconds maximum response time

#### **7.2.2 Microsoft Graph API**

* **Endpoint**: <https://graph.microsoft.com/v1.0>
* **Scopes**: User.Read, Profile
* **Rate Limiting**: Microsoft Graph throttling limits
* **Caching**: 1-hour cache for user profile data

## **8. Security Requirements**

### **8.1 Authentication and Authorization**

* Multi-factor authentication (MFA) support
* Role-based access control (RBAC) implementation
* Token-based authentication with secure storage
* Session timeout after 8 hours of inactivity

### **8.2 Data Protection**

* All data transmission encrypted using TLS 1.3
* Sensitive data encrypted at rest using AES-256
* No persistent storage of conversation content without explicit consent
* PII data handling compliance (GDPR, CCPA)

### **8.3 Infrastructure Security**

* Web Application Firewall (WAF) implementation
* DDoS protection and rate limiting
* Security headers implementation (CSP, HSTS, etc.)
* Regular security vulnerability scanning

### **8.4 Compliance Requirements**

* SOC 2 Type II compliance readiness
* ISO 27001 security controls implementation
* OWASP Top 10 vulnerability mitigation
* Regular penetration testing and security audits

## **9. Performance Requirements**

### **9.1 Response Time Requirements**

* Page load time < 2 seconds (95th percentile)
* AI response initiation < 1 second
* Complete AI response < 10 seconds
* Authentication flow completion < 3 seconds

### **9.2 Scalability Requirements**

* Support 1,000 concurrent users per instance
* Horizontal scaling capability up to 10,000 users
* Auto-scaling based on CPU and memory usage
* Database performance optimization for conversation storage

### **9.3 Availability Requirements**

* 99.9% uptime SLA (8.77 hours downtime/year maximum)
* Graceful degradation during service outages
* Automatic failover for critical components
* Disaster recovery with 4-hour RTO, 1-hour RPO

### **9.4 Resource Optimization**

* Client-side bundle size < 500KB zipped
* Memory usage < 100MB per user session
* CDN utilization for static assets
* Image optimization and lazy loading

## **10. Deployment and Infrastructure**

### **10.1 Development Environment**

* Local development using Next.js dev server
* Docker containerization for consistent environments
* Environment variable management for configuration
* Hot reloading and development tools integration

### **10.2 Staging Environment**

* Production-like staging environment on Azure
* Automated deployment pipeline from development
* Integration testing and end-to-end test execution
* Performance testing and load simulation

### **10.3 Production Environment**

* Azure App Service or Vercel hosting
* Azure CDN for global content delivery
* Azure Application Insights for monitoring
* Automated backup and disaster recovery

### **10.4 CI/CD Pipeline**

# Example GitHub Actions workflow  
name: Deploy to Production  
on:  
 push:  
 branches: [main]  
jobs:  
 test:  
 - Unit tests  
 - Integration tests  
 - Security scanning  
 build:  
 - TypeScript compilation  
 - Bundle optimization  
 - Docker image creation  
 deploy:  
 - Staging deployment  
 - Smoke tests  
 - Production deployment  
 - Health checks

## **11. KPIs and Success Metrics**

### **11.1 User Engagement Metrics**

* **Daily Active Users (DAU)**: Target 70% of registered users
* **Session Duration**: Average 15+ minutes per session
* **Messages per Session**: Average 10+ messages per conversation
* **User Retention**: 80% weekly retention, 60% monthly retention

### **11.2 Performance Metrics**

* **Response Time**: 95% of requests under 2 seconds
* **Error Rate**: < 0.1% application errors
* **Uptime**: 99.9% availability
* **AI Response Quality**: 4.5+ user rating (1-5 scale)

### **11.3 Business Metrics**

* **User Adoption Rate**: 80% of target organization within 30 days
* **Feature Utilization**: 90% of users actively using core features
* **Support Ticket Reduction**: 25% reduction in IT support requests
* **Cost per User**: < $5/month operational cost per active user

### **11.4 Security Metrics**

* **Security Incidents**: Zero critical security incidents
* **Compliance Score**: 100% compliance with organizational requirements
* **Vulnerability Response**: 100% critical vulnerabilities patched within 24 hours
* **Audit Results**: Pass all quarterly security audits

## **12. Timeline**

### **12.1 Phase 1: MVP Development**

**Duration: 12 weeks**

|  |  |  |
| --- | --- | --- |
| **Week** | **Milestone** | **Deliverables** |
| 1-2 | Setup & Authentication | Project setup, Azure AD integration |
| 3-4 | Core Chat Interface | Basic UI, message handling |
| 5-6 | AI Integration | Lab45 API integration, streaming |
| 7-8 | Security & Testing | Security hardening, comprehensive testing |
| 9-10 | Performance Optimization | Performance tuning, optimization |
| 11-12 | Deployment & Launch | Production deployment, monitoring setup |

### **12.2 Phase 2: Enhanced Features**

**Duration: 8 weeks**

|  |  |  |
| --- | --- | --- |
| **Week** | **Milestone** | **Deliverables** |
| 1-2 | Conversation Management | History, search, export features |
| 3-4 | Multi-model Support | Additional AI models, configuration |
| 5-6 | File Upload & Analysis | Document processing capabilities |
| 7-8 | Advanced UI Features | Themes, accessibility improvements |

### **12.3 Phase 3: Enterprise Features**

**Duration: 12 weeks**

|  |  |  |
| --- | --- | --- |
| **Week** | **Milestone** | **Deliverables** |
| 1-3 | Admin Dashboard | Analytics, user management |
| 4-6 | Collaboration Features | Team chat, sharing capabilities |
| 7-9 | Enterprise Integration | LDAP, SAML, compliance tools |
| 10-12 | Scale & Optimize | Performance optimization, scaling |

## **13. Risk Assessment**

### **13.1 Technical Risks**

#### **High Risk**

* **AI Service Dependency**
* ***Risk*:** ChatApp AI service outages affecting core functionality
* ***Mitigation*:** Implement multiple AI provider support, graceful degradation
* ***Probability*:** Medium | *Impact*: High
* **Authentication Integration**
* ***Risk*:** Complex Azure AD integration causing user access issues
* ***Mitigation*:** Extensive testing, fallback authentication methods
* ***Probability*:** Low | *Impact*: High

#### **Medium Risk**

* **Performance at Scale**
* ***Risk*:** Poor performance with high user concurrency
* ***Mitigation*:** Load testing, auto-scaling, performance monitoring
* ***Probability*:** Medium | *Impact*: Medium
* **Security Vulnerabilities**
* ***Risk*:** Security flaws in authentication or data handling
* ***Mitigation*:** Security reviews, penetration testing, compliance audits
* ***Probability*:** Low | *Impact*: High

### **13.2 Business Risks**

#### **High Risk**

* **User Adoption**
* ***Risk*:** Low user adoption due to complex authentication or poor UX
* ***Mitigation*:** User testing, simplified onboarding, training materials
* ***Probability*:** Medium | *Impact*: High

#### **Medium Risk**

* **Competitive Pressure**
* ***Risk*:** Similar solutions from major vendors (Microsoft, Google)
* ***Mitigation*:** Focus on unique enterprise features, rapid iteration
* ***Probability*:** High | *Impact*: Medium

### **13.3 Operational Risks**

* **Team Knowledge Gap**
* ***Risk*:** Limited expertise in AI integration or Azure services
* ***Mitigation*:** Training programs, external consultants, documentation
* ***Probability*:** Medium | *Impact*: Medium
* **Regulatory Changes**
* ***Risk*:** New AI or data privacy regulations affecting product
* ***Mitigation*:** Legal consultation, compliance monitoring, flexible architecture
* ***Probability*:** Low | *Impact*: Medium

## **14. Conclusion**

ChatApp is a strategically sound enterprise AI chat application that effectively combines Azure AD security with modern AI capabilities.

**Assessment**

**Strong Foundation:** Next.js + TypeScript + Azure integration provides robust, scalable architecture

**Clear Roadmap:** Well-defined 3-phase approach (MVP → Enhanced → Enterprise)

**Realistic Targets:** Achievable goals (80% adoption, 99.9% uptime, <2s responses)

**Enterprise-Ready:** Security, compliance, and Microsoft ecosystem integration built-in