Technical Specification: User Authentication System

1. Overview

This document outlines the technical requirements for implementing a user authentication system for the XYZ Web Application. The system will handle user registration, login, password management, and session management with security best practices.

2. Functional Requirements

2.1 User Registration

- Users must be able to create new accounts using email and password
- Email addresses must be unique in the system
- Password must meet complexity requirements:
    - Minimum 8 characters
    - At least one uppercase letter
    - At least one lowercase letter
    - At least one number
    - At least one special character (!@#$%^&*)
- Users must verify their email address before account activation
- Registration form must include: First Name, Last Name, Email, Password, Confirm Password
- Terms of Service acceptance is required

2.2 User Login

- Users can log in using email and password combination
- System must implement account lockout after 5 failed login attempts
- Locked accounts must be unlocked after 15 minutes or by admin intervention
- "Remember Me" functionality should keep users logged in for 30 days
- System must log all login attempts (successful and failed)

2.3 Password Management

- Users can reset passwords using email verification
- Password reset tokens must expire after 1 hour
- Users can change passwords when logged in
- Old password must be verified before setting new password
- Password history: users cannot reuse last 5 passwords
- System must force password change every 90 days for admin users

2.4 Session Management

- User sessions must timeout after 2 hours of inactivity
- Users can have maximum 3 concurrent sessions
- Users can view and terminate active sessions
- Session data must be stored securely

2.5 Multi-Factor Authentication (MFA)

- Optional MFA using TOTP (Time-based One-Time Password)
- Users can enable/disable MFA in account settings
- Backup codes must be provided when MFA is enabled
- MFA required for admin accounts

3. Technical Requirements

3.1 Security Requirements

- All passwords must be hashed using bcrypt with salt rounds ≥ 12
- JWT tokens for API authentication with 1-hour expiration
- HTTPS required for all authentication endpoints
- Rate limiting: 10 login attempts per minute per IP address
- Input validation and sanitization on all user inputs
- SQL injection and XSS protection

3.2 Database Schema

- Users table with fields: id, email, password_hash, first_name, last_name, created_at, updated_at, email_verified, is_active, last_login, failed_login_attempts, locked_until
- Sessions table with fields: id, user_id, token, expires_at, created_at, ip_address, user_agent
- Password_history table with fields: id, user_id, password_hash, created_at

3.3 API Endpoints

- POST /api/auth/register - User registration
- POST /api/auth/login - User login
- POST /api/auth/logout - User logout
- POST /api/auth/forgot-password - Request password reset
- POST /api/auth/reset-password - Reset password with token
- GET /api/auth/verify-email - Email verification
- POST /api/auth/change-password - Change password (authenticated)

- GET /api/auth/sessions - List active sessions
- DELETE /api/auth/sessions/{id} - Terminate session

3.4 Response Formats

- Success responses: HTTP 200 with JSON containing user data and token
- Error responses: HTTP 4xx/5xx with JSON containing error message and code
- Login response must include: access_token, refresh_token, expires_in, user_id

4. Performance Requirements

- Login requests must complete within 2 seconds
- Registration must complete within 3 seconds
- Password reset email must be sent within 30 seconds
- System must support 1000 concurrent users
- Database queries must be optimized with proper indexing

5. Error Handling

- Invalid email format: "Please enter a valid email address"
- Password complexity not met: "Password must meet complexity requirements"
- Account locked: "Account temporarily locked due to multiple failed attempts"
- Invalid credentials: "Invalid email or password"
- Session expired: "Your session has expired. Please log in again"
- Email already exists: "An account with this email already exists"

6. Logging and Monitoring

- Log all authentication events with timestamps
- Monitor failed login attempts and potential security threats
- Track user registration and activation rates
- Alert on unusual login patterns or potential attacks

7. Integration Requirements

- Email service integration for verification and password reset emails
- Integration with existing user profile system
- Single Sign-On (SSO) compatibility for future implementation
- API integration with mobile applications

8. Testing Requirements

- Unit tests for all authentication functions

- Integration tests for API endpoints
- Security testing for vulnerabilities
- Load testing for performance requirements
- End-to-end testing for user workflows

9. Deployment Requirements

- Environment variables for configuration
- Database migration scripts
- SSL certificate configuration
- Backup and recovery procedures
- Monitoring and alerting setup