# Product Requirements Document

# Product Requirements Document: Login Module

## 1. Overview

The Login Module is a critical authentication component that enables users to securely access the application using email and password credentials. This module serves as the primary entry point for authenticated users and must provide a seamless, secure, and user-friendly authentication experience.

## 2. Business Objectives

- Enable secure user authentication with email and passwordĐ
- Provide a frictionless login experience for returning usersĐ
- Implement robust security measures to prevent unauthorized accessĐ
- Support multiple authentication methods (email/password, OAuth)Đ
- Maintain user sessions with appropriate timeout mechanismsĐ

## 3. Functional Requirements

### 3.1 Login Form

FR-001: The login page must display a form with the following fields:

- Email address input field (required)Đ
- Password input field (required, masked)Đ
- "Remember Me" checkbox (optional)Đ
- "Sign In" buttonĐ
- "Forgot Password" linkĐ
- "Sign in with Google" button (OAuth option)Đ

FR-002: Email field validation:

- Must accept valid email format (RFC 5322 compliant)Đ
- Maximum length: 254 charactersĐ
- Must display validation error for invalid formatĐ
- Must display "Email is required" if field is emptyĐ

FR-003: Password field validation:

- Minimum length: 8 charactersĐ
- Must contain at least one uppercase letter, one lowercase letter, one number, and one special characterĐ
- Must display validation error for weak passwordsĐ
- Must display "Password is required" if field is emptyĐ
- Must support special characters: !@#$%^&*()_+-=[]{}|;:,.<>?Đ

FR-004: Form submission:

- Submit button must be disabled until both email and password fields have valid inputĐ
- On submission, display loading indicatorĐ

• On successful login, redirect user to dashboardĐ
• On failed login, display error message without revealing whether email existsĐ
• Error message: "Invalid email or password. Please try again."Đ

### 3.2 Authentication Logic

FR-005: Credential validation:

• System must verify email exists in user databaseĐ
• System must verify password matches stored hash (bcrypt)Đ
• System must check if account is locked or suspendedĐ
• System must check if account requires email verificationĐ

FR-006: Session management:

• On successful login, generate JWT token with 24-hour expirationĐ
• Store token in secure HTTP-only cookieĐ
• Store refresh token in secure HTTP-only cookie (30-day expiration)Đ
• Redirect to originally requested page if user was redirected to loginĐ

FR-007: Remember Me functionality:

• If "Remember Me" is checked, extend session to 30 daysĐ
• Store remember token in secure cookieĐ
• Pre-fill email field on next visit (if remembered)Đ

FR-008: Account lockout:

• Lock account after 5 consecutive failed login attemptsĐ
• Lock duration: 15 minutesĐ
• Display message: "Account locked due to multiple failed attempts. Please try again after 15 minutes."Đ
• Reset lockout counter on successful loginĐ

### 3.3 OAuth Integration

FR-009: Google OAuth:

• Display "Sign in with Google" buttonĐ
• On click, redirect to Google OAuth consent screenĐ
• On successful OAuth, create or link user accountĐ
• Generate session token same as email/password flowĐ
• Handle OAuth errors gracefullyĐ

### 3.4 Security Requirements

FR-010: Password security:

• Passwords must be hashed using bcrypt (cost factor: 12)Đ
• Never store plain text passwordsĐ
• Never return password in API responsesĐ
• Implement password strength meter (visual indicator)Đ

FR-011: Rate limiting:

• Limit login attempts to 5 per minute per IP addressĐ

- Limit login attempts to 10 per hour per email addressĐ
- Return 429 status code with "Too many requests" message when limit exceededĐ

FR-012: Input sanitization:

- Sanitize all user inputs to prevent XSS attacksĐ
- Validate and escape special charactersĐ
- Prevent SQL injection through parameterized queriesĐ
- Block common attack patterns (e.g., <script>, javascript:, data:)Đ

FR-013: HTTPS enforcement:

- All login requests must use HTTPSĐ
- Redirect HTTP requests to HTTPSĐ
- Set Secure flag on all authentication cookiesĐ

## 4. Non-Functional Requirements

### 4.1 Performance

NFR-001: Login response time:

- Page load time: < 2 secondsĐ
- Authentication API response: < 500ms (95th percentile)Đ
- Database query time: < 100msĐ

NFR-002: Concurrent users:

- Support minimum 1000 concurrent login requestsĐ
- Handle traffic spikes during peak hoursĐ

### 4.2 Usability

NFR-003: User experience:

- Form should be accessible via keyboard navigationĐ
- Support screen readers (ARIA labels)Đ
- Display clear, actionable error messagesĐ
- Provide visual feedback for all user actions (loading states, success indicators)Đ

NFR-004: Responsive design:

- Mobile-friendly layout (320px minimum width)Đ
- Tablet-optimized layout (768px - 1024px)Đ
- Desktop layout (1024px+)Đ

### 4.3 Accessibility

NFR-005: WCAG 2.1 AA compliance:

- Color contrast ratio: minimum 4.5:1 for textĐ
- Keyboard navigation supportĐ
- Screen reader compatibilityĐ
- Focus indicators visibleĐ

## 5. User Flows

### 5.1 Successful Login Flow

1. User navigates to login page
2. User enters valid email address
3. User enters valid password
4. User optionally checks "Remember Me"
5. User clicks "Sign In" button
6. System validates credentials
7. System generates session token
8. System redirects user to dashboard
9. Dashboard displays welcome message with user name

### 5.2 Failed Login Flow

1. User navigates to login page
2. User enters email address
3. User enters incorrect password
4. User clicks "Sign In" button
5. System validates credentials (fails)
6. System increments failed attempt counter
7. System displays error message: "Invalid email or password"
8. User remains on login page
9. User can retry with correct credentials

### 5.3 Account Lockout Flow

1. User attempts login with incorrect password (attempt 1)
2. System displays error message
3. User attempts login with incorrect password (attempts 2-5)
4. On 5th failed attempt, system locks account
5. System displays: "Account locked. Please try again after 15 minutes."
6. User must wait 15 minutes before next attempt
7. After 15 minutes, user can attempt login again

## 6. Error Handling

### 6.1 Validation Errors

- **Empty email:** "Email is required"Đ
- **Invalid email format:** "Please enter a valid email address"Đ
- **Empty password:** "Password is required"Đ
- **Weak password:** "Password must be at least 8 characters with uppercase, lowercase, number, and special character"Đ

### 6.2 Authentication Errors

- **Invalid credentials:** "Invalid email or password. Please try again."Đ
- **Account locked:** "Account locked due to multiple failed attempts. Please try again after 15 minutes."Đ
- **Account suspended:** "Your account has been suspended. Please contact support."Đ
- **Email not verified:** "Please verify your email address before logging in. Check your inbox for verification link."Đ

### 6.3 System Errors

- **Server error:** "An error occurred. Please try again later."Đ
- **Network timeout:** "Request timed out. Please check your connection and try again."Đ
- **Rate limit exceeded:** "Too many requests. Please try again in a few minutes."Đ

## 7. Technical Specifications

### 7.1 API Endpoints

POST /api/auth/login

- Request body: `{ "email": "user@example.com", "password": "SecurePass123!", "rememberMe": true }`Đ
- Success response (200): `{ "success": true, "token": "jwt_token", "user": { "id": "123", "email": "user@example.com", "name": "John Doe" } }`Đ
- Error response (401): `{ "error": "Invalid email or password" }`Đ
- Error response (429): `{ "error": "Too many requests", "retryAfter": 60 }`Đ

### 7.2 Database Schema

users table:

- id (UUID, primary key)Đ
- email (VARCHAR(254), unique, indexed)Đ
- password_hash (VARCHAR(255))Đ
- name (VARCHAR(100))Đ
- email_verified (BOOLEAN, default false)Đ
- account_locked (BOOLEAN, default false)Đ
- locked_until (TIMESTAMP, nullable)Đ
- failed_login_attempts (INTEGER, default 0)Đ
- last_login_at (TIMESTAMP, nullable)Đ
- created_at (TIMESTAMP)Đ
- updated_at (TIMESTAMP)Đ

### 7.3 Security Measures

- JWT token signing: RS256 algorithmĐ
- Password hashing: bcrypt (cost factor 12)Đ
- Cookie security: HttpOnly, Secure, SameSite=StrictĐ
- CSRF protection: Token-based validationĐ
- Rate limiting: Redis-based with sliding windowĐ

## 8. Acceptance Criteria

1. ' User can successfully log in with valid email and password
2. ' User receives appropriate error message for invalid credentials
3. ' Account locks after 5 failed login attempts
4. ' "Remember Me" extends session to 30 days
5. ' OAuth login with Google works correctly
6. ' All validation errors display correctly

7. ' Page loads in < 2 seconds
8. ' Form is accessible via keyboard and screen readers
9. ' All security requirements are implemented
10. ' System handles 1000+ concurrent login requests

## 9. Dependencies

- Authentication service (JWT token generation)Đ
- User database (credential verification)Đ
- Email service (verification emails)Đ
- OAuth provider (Google)Đ
- Redis (rate limiting and session storage)Đ
- HTTPS certificate (SSL/TLS)Đ

## 10. Out of Scope

- Password reset functionality (separate module)Đ
- Two-factor authentication (future enhancement)Đ
- Social login with Facebook/Twitter (future enhancement)Đ
- Biometric authentication (future enhancement)Đ
- Single Sign-On (SSO) integration (future enhancement)Đ