**Sprint 3 (5 Days) – Login Functionality**

**1. User Authentication Setup (5 Story Points)**

* **Sub-tasks:**
  + Implement secure password hashing (bcrypt/scrypt).
  + Set up session management (JWT/cookies).
  + Configure user role permissions (admin/user).
  + Integrate rate-limiting for brute-force protection.
  + Log authentication attempts (success/failure).

**2. Login Page Frontend (3 Story Points)**

* **Sub-tasks:**
  + Design responsive HTML/CSS login form.
  + Add client-side validation (email format, password strength).
  + Implement "Remember Me" toggle functionality.
  + Add error message display (invalid credentials, locked account).

**3. Database Integration (3 Story Points)**

* **Sub-tasks:**
  + Connect to user database (SQL/NoSQL).
  + Write queries for credential verification.
  + Optimize DB indexing for login speed.
  + Handle connection errors gracefully.

**4. Backend API for Login (5 Story Points)**

* **Sub-tasks:**
  + Create REST endpoint (POST /login).
  + Validate request payload (email, password).
  + Return proper HTTP status codes (200, 401, 403).
  + Secure API against SQL injection & XSS.

**5. Error Handling & Security (2 Story Points)**

* **Sub-tasks:**
  + Display user-friendly error messages.
  + Implement account lockout after 5 failed attempts.
  + Log security-related events (e.g., multiple failed logins).

**6. Testing & QA (2 Story Points)**

* **Sub-tasks:**
  + Unit tests for auth functions (Jest/Pytest).
  + End-to-end testing (Cypress/Selenium).
  + Security audit (OWASP ZAP penetration test).

**Total Story Points: 20**

**Key Dependencies:**

* User database schema must be finalized.
* DevOps ready for deployment (CI/CD pipeline).

**Acceptance Criteria:**

* Users can log in securely with email & password.
* Failed logins trigger appropriate warnings/lockouts.
* All edge cases (empty fields, invalid inputs) handled.