P4 Manual_Log

Features Implemented:

Frontend:

- Login component with username and password fields
- Register component with password confirmation
- Main app shows the logged-in username with an admin badge if admin
- Article posting form with URL and optional title fields
- Article list displaying all posts with author, timestamp, and clickable links
- Delete button appears only on own articles (or all articles for the admin)
- Logout button in the header
- JWT token stored in localStorage for persistent login
- Axios is configured to send a token in the Authorization header
- React hooks (useState, useEffect) manage authentication state
- Token automatically decoded on page load to restore the session
- Error messages are displayed for failed operations

Backend:

- Express.js server on port 5001
- SQLite database initialized in database.js
- Auth routes in routes/auth.js: POST /api/auth/login and POST /api/auth/register
- Article routes in routes/articles.js: GET /api/articles, POST /api/articles, DELETE /api/articles/:id
- JWT middleware in middleware/auth.js verifies tokens - Bcrypt with 10 rounds hashes passwords
- Admin user auto-created with username "admin" and password "admin"
- express-validator checks username length (3+ chars), password length (6+ chars), and URL format
- CORS allows localhost:3000 and localhost:3001
- Helmet adds security headers
- Rate limiter set to 100 requests per 15 minutes
- Parameterized SQL queries throughout (db.run with ? placeholders)

Security (OWASP Top 10):

- A01:2025 - Broken Access Control
  - JWT tokens required for all article operations, users can only delete own posts, admin can delete any post, protected routes with authenticateToken middleware

- A02:2025 - Security Misconfiguration
  - Helmet.js sets security headers (X-Content-Type-Options, X-Frame-Options, etc.), CORS restricted to specific origins (localhost:3000, localhost:3001), generic error messages don't leak implementation details

- A04:2025 - Cryptographic Failures
  - Bcrypt hashes passwords with 10 salt rounds, JWT tokens use secret key for signing, no plaintext passwords stored

- A05:2025 - Injection
  - All database queries use parameterized statements with ? placeholders (prepare().get(), prepare().run()), no SQL string concatenation, express-validator escape() sanitizes inputs

- A07:2025 - Authentication Failures
  - Passwords must be 6+ characters minimum, usernames must be 3+ characters, JWT tokens expire in 24 hours, rate limiting (100 req/15min) protects against brute force, generic error messages for failed auth

- A10:2025 - Mishandling of Exceptional Conditions
  - Try/catch blocks on all route handlers, error handling middleware catches unhandled errors, proper HTTP status codes (400, 401, 403, 404, 500), database errors logged server-side

Features Not Implemented:

- A03:2025 - Software Supply Chain Failures
- A06:2025 - Insecure Design
- A08:2025 - Software or Data Integrity Failures
- A09:2025 - Logging & Alerting Failures (only basic console logging implemented)

- When logged in on Manual and an article is posted, the site must be refreshed for the article to appear on the site.