

P4 Manual_Log

Features Implemented:

- Frontend:
 - Fully functional Login page to allow users to log in with credentials.
 - Fully functional Register page to allow users to create accounts.
 - Comprehensive Dashboard page that displays user's name and all shared articles. The Dashboard allows the creation of new article posts, deletion of article posts, and logout functionality.
 - Authorization functionality using React Context (AuthContext.js) to handle auth state across the app.
 - JWT token storage in localStorage for persistent sessions.
 - Protected routes that redirect unauthenticated users to the login page if JWT token is not verified.
 - Automatic token verification on app load to restore user sessions.
- Backend:
 - Express.js server that utilizes REST for API calls to the frontend.
 - SQLite database for user credential and article post storage.
 - User authentication API endpoints for login, register, and me (get user info).
 - Article API endpoints: GET articles, POST articles, and DELETE articles (if user is owner of post or admin).
 - JWT authentication with token utilization (24hr expiration).
 - Bcrypt password hashing with 12 salt rounds.
 - Middleware for token verification and admin auth
 - Validation for username length, password length, and proper URL formatting.
- Security (OWASP Top 10):
 - A01:2025 - Broken Access Control
 - JWT, Auth, Protected Routes
 - A02:2025 - Security Misconfiguration
 - Helmet.js
 - A04:2025 - Cryptographic Failures
 - Bcrypt hashing with 12 salt rounds
 - A05:2025 - Injection
 - Use indirect SQL commands with ? as placeholders

- A07:2025 - Authentication Failures
 - Rate limiting, hashing, error messages
- A10:2025 - Mishandling of Exceptional Conditions
 - Try/catch blocks on all route handlers

Features Not Implemented:

- Security (OWASP Top 10):
 - A03:2025 - Software Supply Chain Failures
 - A06:2025 - Insecure Design
 - A08:2025 - Software or Data Integrity Failures
 - A09:2025 - Logging & Alerting Failures