P4 Vibe_Log

Features Implemented:

Frontend:

- Single HTML page in public/index.html with embedded CSS and JavaScript
- Login form and register form that toggle visibility
- Shows username and admin badge when logged in
- Article posting form with URL and optional title inputs
- Article list showing all posts with delete buttons
- Delete buttons visible only for own posts (admin sees them on all posts)
- Logout button in header
- JWT token stored in localStorage
- Fetch API sends token in Authorization header
- JavaScript decodes JWT to get user info on page load
- Error messages show in red boxes and auto-dismiss after 5 seconds
- Success messages in green boxes
- Purple to blue gradient background with card-style layout
- Responsive design

Backend:

- Single index.js file contains entire Express server
- SQLite database created with users and articles tables
- Routes defined inline: POST /api/login, POST /api/register, GET /api/articles, POST /api/articles, DELETE /api/articles/:id
- JWT tokens created with jsonwebtoken, expire in 24 hours
- Bcryptjs hashes passwords with 10 rounds (using hashSync and compareSync)
- Admin user auto-created on startup if doesn't exist (username: admin, password: admin)
- Token verification middleware checks Authorization header on protected routes
- Only article owner or admin can delete articles
- Basic input validation: username 3+ chars, password 6+ chars, URL must match http/https pattern
- CORS enabled for all origins
- Parameterized SQL queries using db.run and db.get with ? placeholders
- Error handling with try/catch blocks

Security (OWASP Top 10 2025):

- A01:2025 - Broken Access Control
    - JWT authentication required for protected routes, authenticateToken middleware verifies tokens, role-based deletion permissions (user can delete own, admin can delete any)

- A04:2025 - Cryptographic Failures
    - Bcryptjs hashing with 10 rounds, JWT secret key for token signing, no plaintext passwords

- A05:2025 - Injection
    - Parameterized queries with ? placeholders (db.run, db.get), no SQL string concatenation

- A07:2025 - Authentication Failures
    - Password length validation (6+ chars minimum), username validation (3+ chars), token expiration (24 hours)

- A10:2025 - Mishandling of Exceptional Conditions
    - Try/catch blocks on authentication routes, database error handling, proper HTTP status codes

Features Not Implemented:

All required features completed within free AI tokens.

Not included (simpler implementation compared to manual):
A02:2025 - Security Misconfiguration (basic CORS only, no Helmet.js)
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)
Rate limiting (manual version has this)
Helmet.js security headers (manual version has this)
express-validator sanitization (manual version has this)