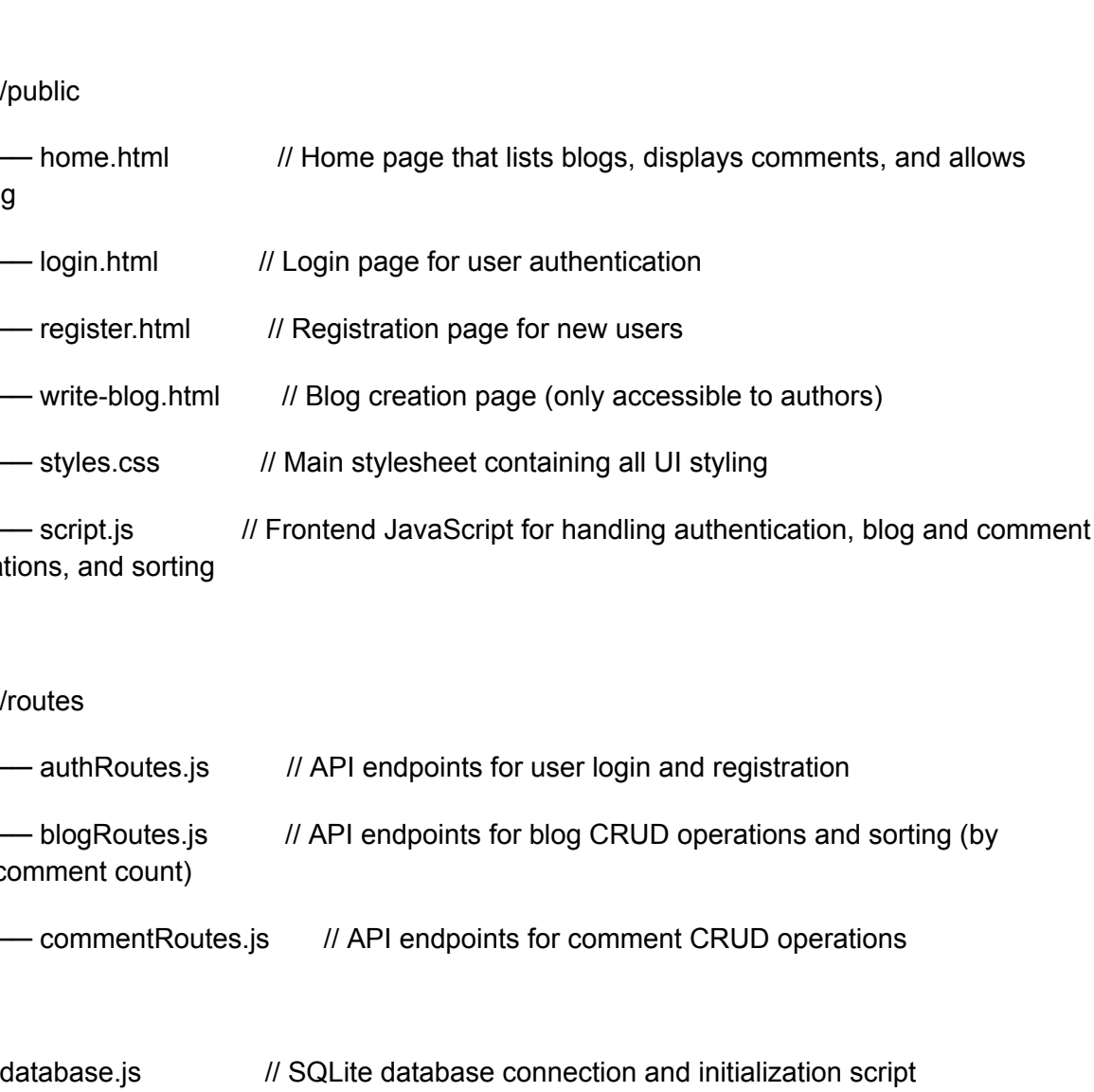
Blog Management System Documentation

Project Overview

The project is a fully functional Web-Based Blog Management System supporting user authentication, blog creation with full CRUD operations, and a sorting system with sorting features. This snapshot represents the project folder state immediately after successful testing.

Folder Structure

/Blog-Management-System



├── server.js // Express server setup and configuration (loads environment variables via dotenv)

├── package.json // Project manifest listing dependencies (express, cors, dotenv, bcryptjs, jsonwebtoken, sqlite3 or better-sqlite3)

├── package-lock.json // Auto-generated file that locks dependency versions

└── .env // Environment variables file (contains JWT_SECRET, PORT, etc.)

Detailed Description

1. /public Folder

- **HTML Files:**
  - **home.html:** Displays the list of blogs, sorting buttons, and comment sections.
  - **login.html & register.html:** Provide user authentication forms.
  - **write-blog.html:** Contains a form for authors to create new blog posts.
- **styles.css:**
  - Contains the UI styling with a chosen color palette and modern design.
- **script.js:**
  - Handles frontend functionality including authentication, blog and comment CRUD operations, and sorting features.

2. /routes Folder

- **authRoutes.js:**
  - Implements the API endpoints for user registration and login, utilizing bcryptjs for hashing passwords and jsonwebtoken for generating tokens.
- **blogRoutes.js:**
  - Provides endpoints to create, read, update, and delete blogs.
  - Implements sorting functionality by time and by the number of comments.
- **commentRoutes.js:**
  - Offers endpoints for adding, editing, and deleting comments associated with blogs.

3. Backend Core Files

- **database.js:**

    - Establishes a connection to the SQLite database, initializes the required tables (Users, Blogs, Comments), and ensures the database is set up for the application.
- **server.js:**
    - Sets up the Express server, configures middleware (CORS, JSON parsing, static file serving), loads environment variables using dotenv, and mounts the API routes.

4. Configuration Files

- **package.json & package-lock.json:**
    - List all dependencies required for the application (e.g., Express, bcryptjs, jsonwebtoken, SQLite3 or better-sqlite3, dotenv, cors).
- **.env:**
    - Contains sensitive configurations such as `JWT_SECRET` and `PORT`, which are used by the server at runtime.

Testing Summary

- **User Authentication:**
    - Registration and login have been tested and are working correctly. Users can register and then log in with valid credentials.

- **Blog Operations:**
    - Blog CRUD operations (create, read, update, delete) have been implemented and verified through both the frontend and API testing tools (Postman/cURL)
      .
- **Commenting System:**
    - Comments can be added, edited, and deleted. The comment count is correctly displayed for each blog.

- **Sorting Functionality:**
    - Blogs can be sorted by time and by the number of comments. Toggle buttons on the home page allow users to switch sorting order.

- **Deployment:**
    - The backend is deployed on Railway with proper configuration, and the frontend is hosted on GitHub Pages with working API integration.