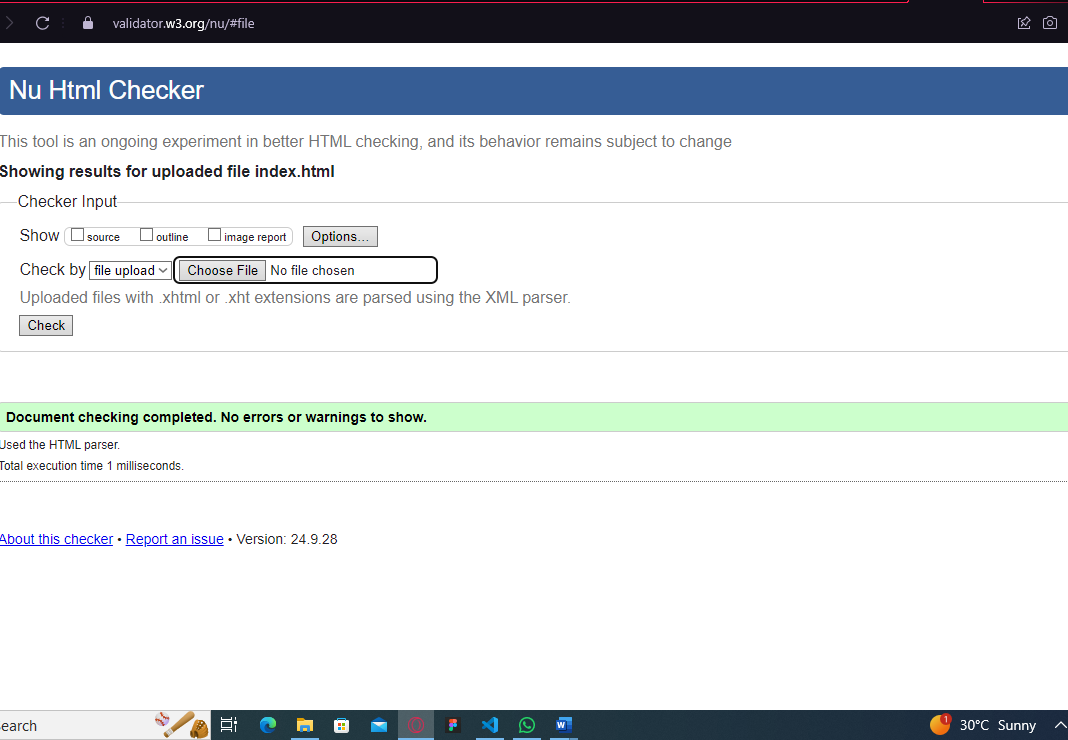
1



2

The database is managed at the server-side using SQLite3.

- Server-side: The Express.js server connects to the SQLite database (`themePark.db`), which stores two tables: `Areas` (themed park areas) and `Attractions` (rides/attractions linked to areas). When a client requests information (like viewing themed areas or attractions), the server queries the database, retrieves the data, and passes it to EJS templates for rendering.

- Client-side: The client (user) interacts with the website through their browser by sending requests (e.g., visiting different pages or submitting forms). These interactions trigger requests to the server, but the actual database operations (data retrieval, updates) happen on the server. The client only sees the rendered HTML content (via EJS) that the server sends back based on the data retrieved from the database.

Thus, the database operations and business logic remain on the server-side, while the client-side handles user interactions and displays the content.

3.

Legal and Ethical Considerations

When developing the marketing website for the theme park, legal considerations were addressed by sourcing images exclusively from Unsplash. This platform provides high-quality, royalty-free images that can be used for commercial purposes without the need for attribution, ensuring compliance with copyright laws and ethical standards.

### Accessibility Considerations

To enhance accessibility, all images on the website have been provided with descriptive alt text. This practice allows visually impaired users to understand the content and context of the images through screen readers. By ensuring that the website meets accessibility standards, we promote inclusivity and ensure that all users can effectively engage with the content.

4.

Security Considerations

1. Use of HTTPS: Although not explicitly mentioned in the code, it is essential to serve the website over HTTPS. This encrypts data transmitted between the client and server, protecting sensitive information from being intercepted by malicious actors.

2. Error Handling: The server includes error messages for database retrieval issues (e.g., `res.status(500).send('Error retrieving areas');`). While helpful for debugging during development, these messages should be generic in a production environment to avoid exposing sensitive information about the database structure or server configuration.

3. Database Security: The SQLite database is secured by ensuring proper permissions and avoiding direct access to the database files. Limiting database access to the application helps protect against unauthorized access.

5.

ERD diagram for the database

