Crazy Car Rental Application Report – Fergal Hughes 20054847

The system is a full-stack web application developed using React, Node js, Express, Bootstrap, and MySQL. The aim of the system is to allow customers to register, log in, browse available cars and make rental bookings through a simple user interface.

This application flows a client-server architecture. The front end is built using React and React Router to enable multipage experience without page reloads. Communication between the front end and the back end is handled with Axios and Restful API endpoints. The back end is built with Node js and Express, which manages the logic, validation and database interactions. A MySQL database is used to store all the persistent data.

See ERD folder.

My approach

The database was designed first to ensure that entities and relationships were clearly defined before any application logic was implemented. This helped me establish a strong structure and reduce issues later. Primary keys and foreign keys were planned to enforce relationships between customers, cars and rentals to ensure good data integrity.

The application follows a clear separation of concerns. The front end is built using React and is responsible for the presentation, interaction, and basic client-side validation. It does not contain any business logic or direct database access. The back end developed using Node js and express handles all the business logic, sever-side validation, and communication with the database. This separation improves maintainability, scalability, and security.

The overall architecture follows the client-server model. React communicates with the Express server using Axios to send HTTP requests to RESTful Api endpoints. Express process these requests, performs validation, and executes queries using the database. The database returns results to the server, which are then sent back to the React front end as a JSON response. This structure provides a clear flow of the data from the user to the database and back.