Design Patterns in FamCamp RV Park Project

Created by the Tech Titans

The design patterns included in the FamCamp RV Park project include the Facade structural pattern and the Chain of Responsibility behavioral pattern. Following are explanations of how these patterns are used in the program.

Facade (Structural Pattern)

In the database file, we use the third party mysql2 package to connect and execute to the database. Instead of sharing the connection throughout the site for calling stored procedures, we have created exportable JavaScript interface methods that call the stored procedures for us. All we have to do is pass in parameters. From the perspective of the programmer that is coding in any of the routes, all they need to do is grab an instance of the database “class” and call the method. The person programming in the routes does not need to know all of the ins and outs of mysql2, they just need to know which parameters to pass to the interface that calls the stored procedure for them. In other words, this “provides a simple interface to a complex subsystem which contains lots of moving parts” (refactoring.guru).

Chain of Responsibility (Behavioral Pattern)

According to refactoring.guru, in the Chain of Responsibility design pattern, handlers are set up in a chain to process requests. “Upon receiving a request, each handler decides to either process the request or to pass it to the next handler in the chain.” The “handlers” in this program are the routers. Based on the input entered on the page by the user, the router will either process the data, reroute to a different page for the processing of the data, or terminate the processing if it lacks the data required. For example, if a user is viewing the page while not logged in and tries to reserve a campsite, the router will reject the request to process the data and reroute to the login page. However, if the user is logged in, the request will move down the chain and be processed to create the reservation.