CS 499

Gene Bryant  
Charles Morris, Jr.

23 March 2025

Module Three Narrative – Software Design and Engineering

In my previous course, CS 465: Full Stack Development I, I took a static vacation booking website and turned the Travel section into a full-stack application using the MEAN stack. On the customer-facing side, users can view the travel packages that are available to them. These packages are pulled from a database and rendered into a web page for the user to view. A single-page application allows administrators to add, modify, and remove available travel packages. I decided to change the Room section of the website into a full-stack application similar to the Travel section.

For this enhancement, I began by working on the customer-facing website. I changed the *rooms.html* file into a Handlebars template and added JavaScript files in the *routes* and *controllers* folders to allow Express to serve the file correctly. The process was relatively straightforward aside from one problem. The *header.hbs* file always styled the “Travel” menu heading as “selected,” even if I was on the “Rooms” page. I had to add code to *rooms.js* and *travel.js* in the *controllers* folder to pass an object named *currentPage* to header.hbs to designate which menu heading was selected.

The next part of the enhancement involved adding a new page to the single-page application to allow administrators to modify the available rooms to choose from. I had to create the Angular components *add-room, edit-room,* and *rooms*. These components are analogous to existing components for displaying, adding, and editing trips. I also added a new link on the navbar so administrators could switch between viewing trips and rooms.

Currently, the room data is stored as JSON *rooms.json* and *rooms.ts* files. Because the database is not yet set up for the room data, I did not write the code necessary to allow administrators to add and update rooms in the single-page application. This is why the “Add” and “Update” buttons are currently disabled. In the future, I will update the code and database to store the room data in the MongoDB database, like the trip data. This will allow the customer-facing website and the single-page application to pull room information from the database.

In addition, I decided to make a couple of minor additions to my original plan. First, I updated the styling of the navbar to make it more readable. I also added cancel buttons to the forms for adding and editing trips and rooms.