Travlr GetAways was the perfect artifact to use in my portfolio as it was a recently developed project that needed enhancements in all three areas in order to bring it closer to meeting client wireframe requirements. I am passionate about web development and using this artifact showcases many skills including front end development, back end development, MEAN stack development, and use of multiple languages and frameworks. Specifically for this portion of enhancements, I implemented CRUD methods to the admin database so that admins are able to add, edit, and delete rooms on both the admin Angular SPA and the customer facing Express web page. Adding in and testing the CRUD methods showcases the following skills: develop front and back end web pages with client requirements prioritized, develop code with a security based mindset, design a database, and optimize a database.

See screenshots for CRUD method functionality testing procedures and results and for further information on security measures in place.

After logging in with authorized credentials I added the following “Test Room” by clicking on the “Add Room” button and filling out all required fields.

A screenshot of a computer

Description automatically generated

After pressing save you can see the new “Test Room” now renders on the admin site and on the customer facing site as seen below:

Admin site:

A screenshot of a computer

Description automatically generated

Customer facing site:

A screenshot of a computer

Description automatically generated

Let’s edit the image on the room to test our EDIT method :

I edited the image from “rooms.png” to “first-class.jpg”

A screenshot of a computer

Description automatically generated

After pressing save the newly updated room now renders on the admin site as so :

A screenshot of a computer

Description automatically generated

And on the customer facing site as below:

A screenshot of a computer

Description automatically generated

Let’s test our final CRUD operation, which is the DELETE operation. We’ll delete the Test Room we’ve entered in the database by clicking the DELETE button listed at the bottom of the room card.

Screenshot of a room with a window and curtains

Description automatically generated

After pressing DELETE , the following page renders on the admin site :

A screenshot of a computer

Description automatically generated

And when checking the rooms-list on the admin page you can see that the Test Room has successfully been removed from the database:

A screenshot of a web page

Description automatically generated

And when checking the customer facing page, we can see that the TEST ROOM has been removed from the database there as well. See below :

A screenshot of a computer

Description automatically generated

Did you meet the course objectives you planned to meet with this enhancement in Module One? Do you have any updates to your outcome-coverage plans?

 I did meet the course objectives planned for this enhancement in Module One. No updates to my outcome-coverage plans. The following outcomes were met in this enhancement :

Demonstrate an ability to use well-founded and innovative techniques, skills, and tools in computing practices for the purpose of implementing computer solutions that deliver value and accomplish industry-specific goals (software engineering/design/database)

By implementing CRUD methods for admins so that admins can make real time changes to the database that appear on the customer facing page immediately.

Develop a security mindset that anticipates adversarial exploits in software architecture and designs to expose potential vulnerabilities, mitigate design flaws, and ensure privacy and enhanced security of data and resources

By implementing Passport in Express so that only authorized users are able to make changes to the data base

An example of the authentication requirements can be found here :

I first integrated Express passport in the \travlr\app\_api\config\passport.js file

A screenshot of a computer program

Description automatically generated

After configuring Passport to validate user name and password credentials I added further security requirements in travlr\app\_api\routes\rooms.js as can be seen below. Notice ‘auth’ before the post, put, and delete functions. This requires authorization from Passport prior to being able to perform each function.

A screenshot of a computer code

Description automatically generated

Furthermore, if an authorized user is not logged into the admin site the Edit,Add, and Delete buttons do not even appear on the room-cards. This can be seen below :

A screenshot of a computer

Description automatically generated

For this enhancement the following files were added and/or modified :

travlr\app\_api\controllers\rooms.js - added roomsAddRoom,roomsUpdateRoom, and roomsDeleteRoom functions

\travlr\app\_api\routes\rooms.js - added lines 10-12 and added auth requirements to each function

\travlr\travlr-admin\src\app\add-room\add-room.component.ts - completed the add room component on the travlr-admin page

\travlr\travlr-admin\src\app\edit-room - completed the edit room component on the travlr-admin page

\travlr\travlr-admin\src\app\delete-room - completed the delete room component on the travlr-admin page

Reflect on the process of enhancing and modifying the artifact. What did you learn as you were creating it and improving it? What challenges did you face?

Modifying the artifact in this enhancement was rewarding as I got to bring together the previous two enhancements with the database to create a completely finished artifact. Now that all three artifacts are finished admins can now add,edit and delete rooms in the database and those changes are immediately reflected on the customer facing page. This portion of enhancements brings the final project one step closer to completion. These enhancements would need to be repeated a few more times with news, about, contact us, etc.. , a shopping cart, login, and register method needs to be implemented on the customer facing page before the project meets client wireframe requirements. Replicating the MVC route and CRUD functionality for contact,about,meals and rooms portions of the page should be fairly simple as I created code that was easy to replicate and re-use when and if needed. The challenges I faced in this enhancement were mostly just related to adding authentication to the add,edit and delete functions for rooms. I kept getting an error that “Get User” is not defined and at that point I realized that I had forgotten to add the Get User function to the \travlr\app\_api\controllers\rooms.js file. After adding the getUser function, which I was able to re-use from the \travlr\app\_api\controllers\trips.js file , the error was resolved. At that point I was able to start testing and ensuring that all functions were working in the way that they were intended to work. Adding CRUD functionality to Rooms enhanced the Travlr database and integrated the software and design enhancement made in enhancement one with the data structure enhancement made in enhancement two.