



Travlr Getaways Full Stack Application
CS 465 Project Software Design Document
Version 1.2

Table of Contents

CS 465 Project Software Design Document	1
Table of Contents	2
Document Revision History	2
Instructions	Error! Bookmark not defined.
Executive Summary	3
Design Constraints	3
System Architecture View	3
Component Diagram	3
Sequence Diagram	4
Class Diagram	5
API Endpoints	6
The User Interface	7

Document Revision History

Version	Date	Author	Comments
1.0	03/24/24	Joseph Veneski	Completed Executive Summary, Design Constraints, and System Architecture View (component diagram) sections.
1.1	04/07/24	Joseph Veneski	Sequence and Class Diagrams, current API Endpoints
1.2	04/21/24	Joseph Veneski	User Interface

Executive Summary

The Travlr Getaways web application uses the MEAN stack, comprised of MongoDB, Express.js, Angular, and Node.js. The website uses a static front-end for handling customer interactions with a dynamic, interactive SPA (Single Page Application) for administrative functions.

The customer-facing front-end uses Node.js and Express.js to render static HTML content. Handlebars templating engine helps streamline design by injecting JSON into HTML for dynamic content delivery. The backend database supplying content is MongoDB, a flexible non-relational structure which is robust and scalable.

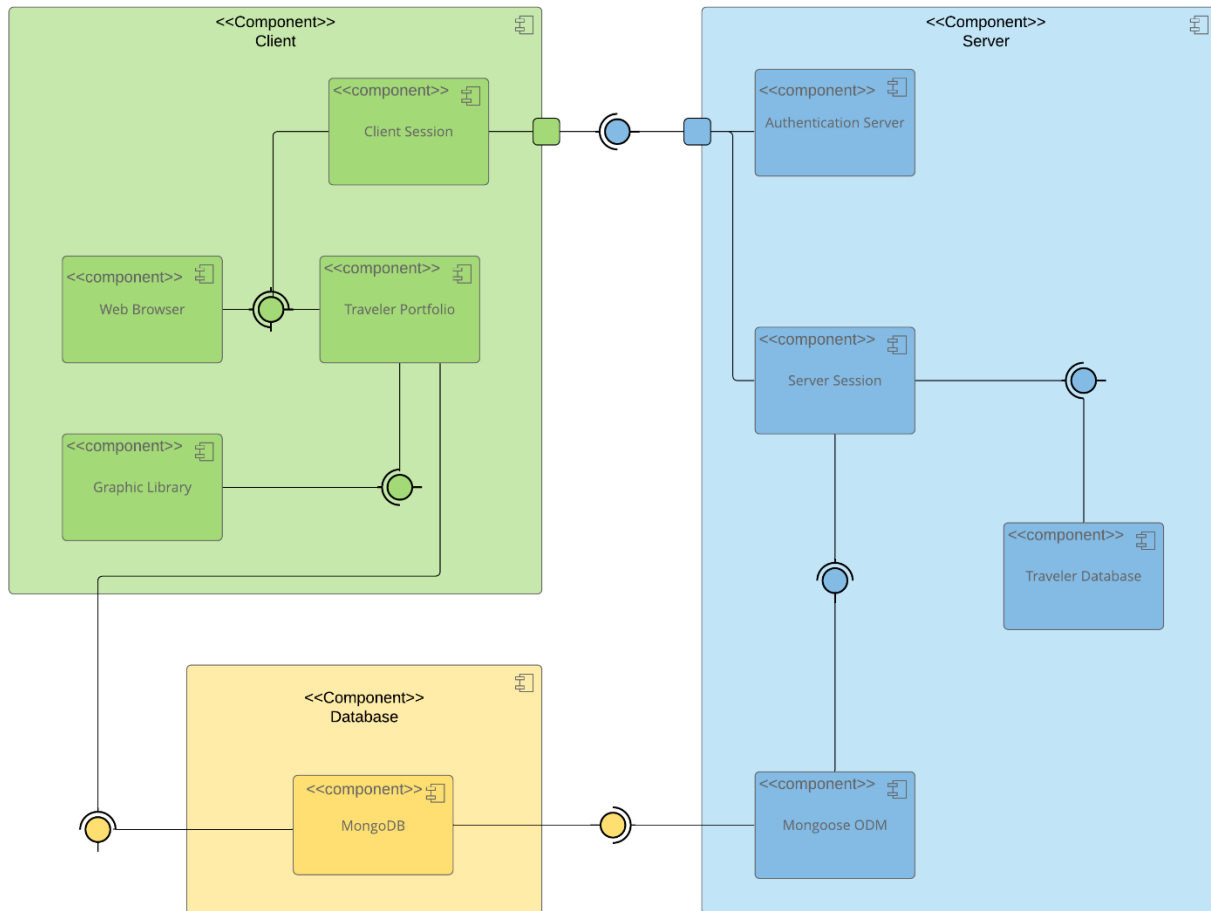
The administration portion of the Travlr web application is built using Angular as a SPA. The Angular framework allows for administrators to easily manage content without needing to refresh the page. To ensure system security, the SPA implements a user authentication process. Administrators are required to log in before they can manage content and user profiles for the Travlr Getaways application. The combination of Angular and Node.js offers a seamless and easy experience for managing the server data through a custom REST API.

Design Constraints

Utilizing the MEAN stack allows the entire application to use JavaScript, however, limits the available third-party tools to ones that are compatible with JavaScript. Use of MongoDB as a schema-less database solution has benefits and limitations. MongoDB is flexible and scalable, however requires careful management of data structure to maintain data integrity and performance. Use of Angular for a SPA approach may limit SEO (Search Engine Optimization) due to not being able to index dynamically loaded content. The SPA's use of dynamically loaded content may be impaired with poor internet connectivity resulting in slow load times and frustration.

System Architecture View

Component Diagram



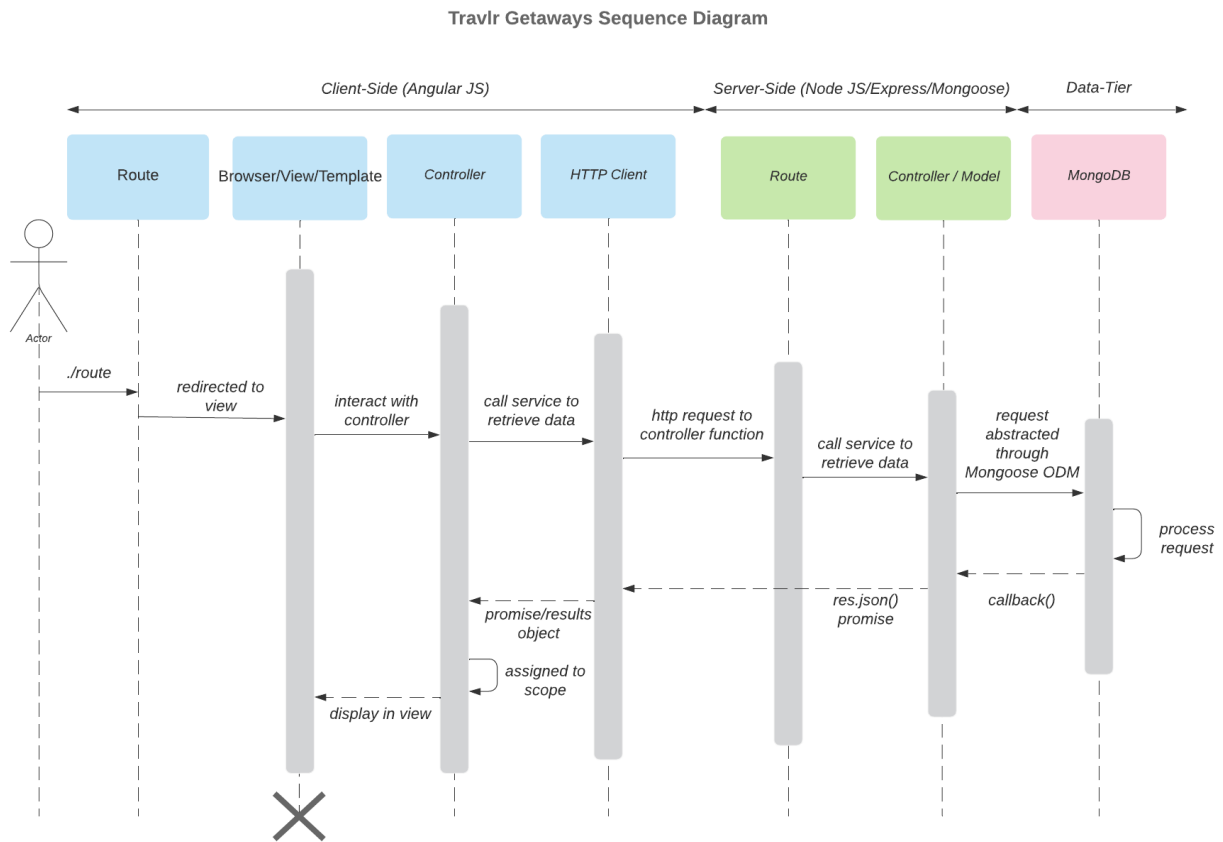
A text version of the component diagram is available: [CS 465 Full Stack Component Diagram Text Version](#).

The component diagram above shows how the Travlr Getaway web application is broken down into three core components, the Client, Server, and Database. The Client uses a web browser for user interactions with a graphic library for rendering visual elements. It handles client sessions maintaining the state between the web browser and server. The Traveler Portfolio handles personalized user data supplied from the Database.

The Server contains the Authentication Server for ensuring secure access to the application. The Server components work together to manage requests from the Client side and interact with the Database. Mongoose ODM interacts directly with the database to model application data. Server sessions are managed and kept in sync with Client Sessions.

The Database component uses MongoDB for a flexible and scalable schema-less solution. Mongoose ODM from the Server side facilitates database interactions. The three main components comprise the Travlr Getaway web application for a secure and seamless experience.

Sequence Diagram



When the user (Actor) enters a URL into a web browser, the AngularJS frontend framework captures this as a 'Route'. The route directs the web browser to load the appropriate View using Handlebars templates. The Angular Controller is called which uses an HTTP Client service to make a request of the server.

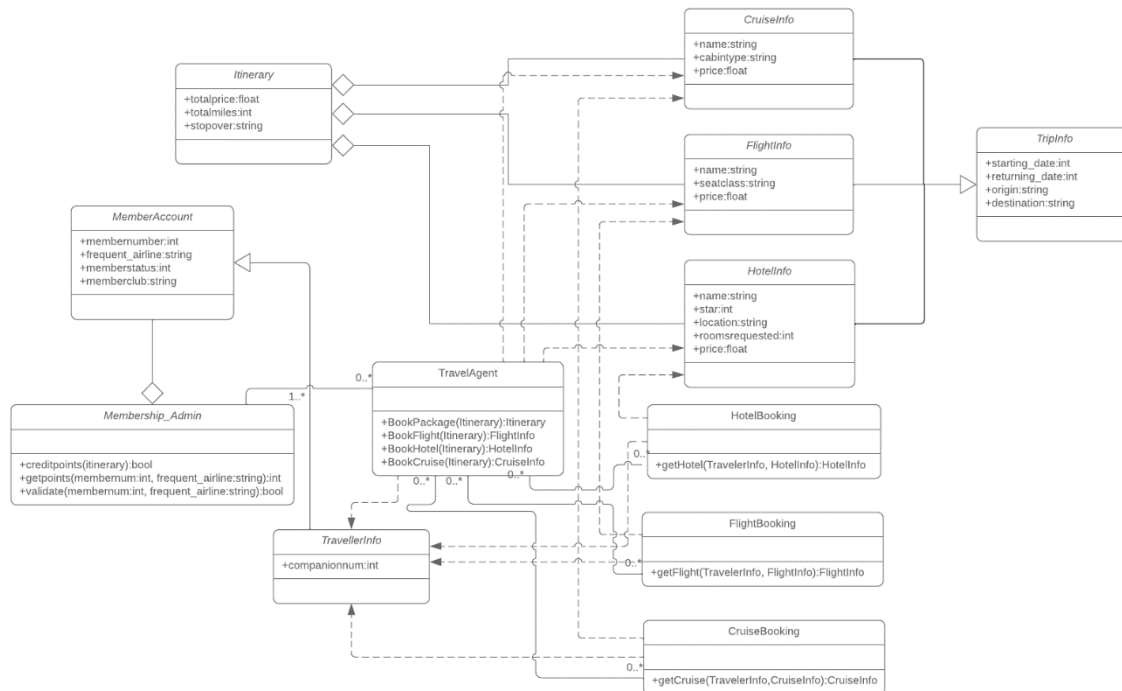
The request is routed through the Express backend, where a delegated Controller / Model is then called. The backend controller processes the request using Mongoose, an Object Data Modeling tool, to abstract the request and interact with MongoDB.

Once MongoDB processes the query, the result is returned to the backend Controller, which then sends the data along with a response status back to the HTTP Client through the Express Route. The Angular controller assigns the returned data to scope, allowing it to be dynamically rendered in the View for the user.

To illustrate the process, normal operations such as Logging in can be described. Users will be prompted to enter their credentials into the View which are sent to the backend through Angular's HTTP Client. The Express backend verifies the credentials against the records in MongoDB. Whether the credentials are verified as valid or not, a response is generated and passed back to the frontend using the Angular Controller to update the View, providing feedback to the user.

Class Diagram

Travlr Getaways Class Diagram



The Itinerary class aggregates trip data from CruiseInfo, FlightInfo, and HotelInfo classes. These three classes are all related to a common class, TripInfo which contains attributes such as start and return dates, origin, and destination. The Info classes themselves each have unique attributes specific to their mode of transport as well as prices.

The Membership_Admin class is used to perform account management operations using data from the MemberAccount class which holds information such as member account numbers, preferred airline, status, and clubs.

The TravellerInfo class extends the MemberAccount class inheriting attributes and adding a companion number. This class has an association with the TravelAgent class as well as the HotelBooking, FlightBooking, and CruiseBooking classes to aid in booking operations.

The TravelAgent class which has zero-to-many relationships with the TravellerInfo and different booking classes helps guide the overall arrangement of travel services. This class also has connections back to the Info classes through dependencies linking the whole system of interactions back to the Itinerary class which aggregates traveler and trip information facilitated by a travel agent.

API Endpoints

Method	Purpose	URL	Notes
GET	Retrieve list of all Trips	/api/trips	Returns all trips in database
GET	Retrieve single trip by code	/api/trips/:tripCode	Returns single trip, identified by passed parameter tripCode
PUT	Update a single trip by code	/api/trips/:tripCode	Updates a single trip identified by tripCode
DELETE	Deletes a single trip by code	/api/trips/:tripCode	Deletes a single trip identified by tripCode
GET	Retrieve list of all Meals	/api/meals	Returns all current meals listed in database
GET	Retrieve single meal by code	/api/meals/:mealCode	Added codes for all meals to simplify. Returns single matching meal by code
GET	Retrieve list of all Rooms	/api/rooms	Returns all rooms in the database
GET	Retrieve a single room by code	/api/rooms/:roomCode	Added codes for all rooms to simplify. Returns single matching room by code

The User Interface

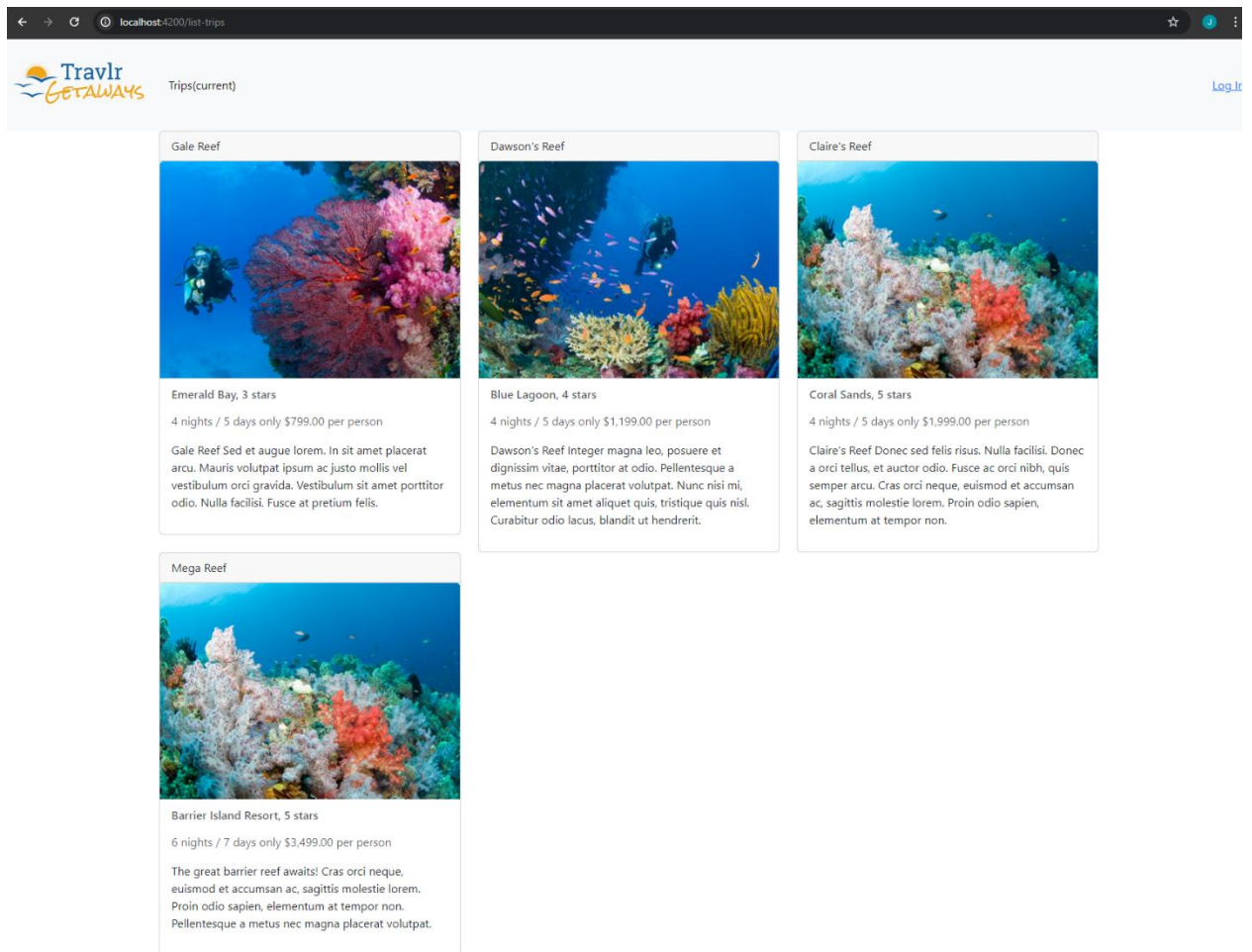


Figure 1 - Logged out view without access to add/edit/delete.

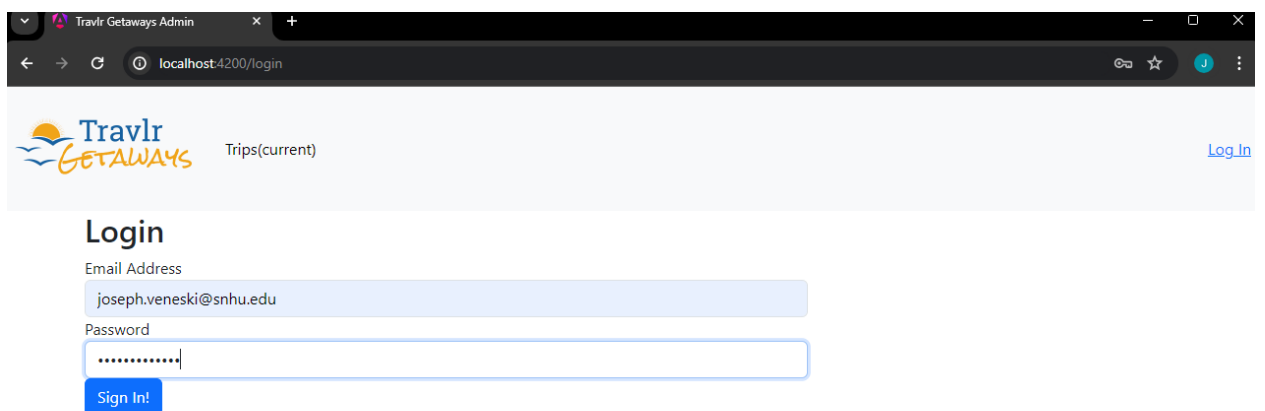


Figure 2 - User authentication.

Travlr Getaways Admin localhost:4200/add-trip Log Out

Travlr
GETAWAYS Trips(current)

Add Trip

Code:
BAHA

Name:
Bahamas

Length:
4 night / 3 day

Start:
04/21/2024

Resort:
Bahamas Resort, 4 star

Per Person:
350.00

Image Name:
reef2.jpg

Description:
Fake Trip added for testing purposes. Testing test

Save

Figure 3 - Adding a custom trip to the database.

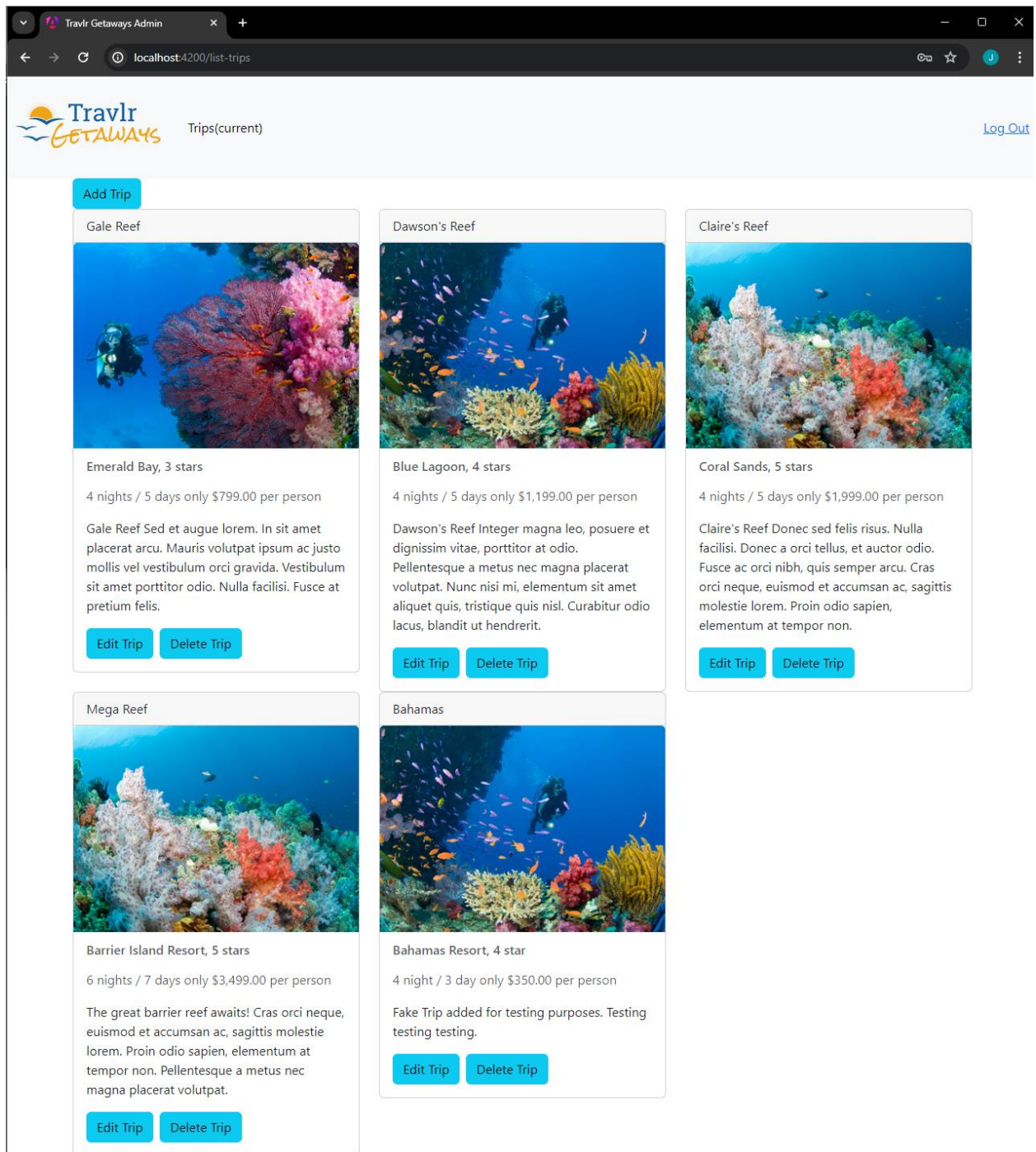


Figure 4 - Custom trip added, all buttons active when logged in.

Travlr Getaways Admin

localhost:4200/edit-trip

Travlr GETAWAYS Trips(current) [Log Out](#)

Edit Trip

Code:
BAHA

Name:
Bahamas

Length:
4 night / 3 day

Start:
04/20/2024

Resort:
Bahamas Resort, 4 star

Per Person:
1350.00

Image Name:
reef2.jpg

Description:
Fake Trip added for testing purposes. Testing test

Save

Figure 5 - Editing the custom trip.

Travlr Getaways Admin

localhost:4200/delete-trip

Travlr
GETAWAYS

Trips(current)

[Log Out](#)

Delete Trip

Code:
BAHA

Name:
Bahamas

Length:
4 night / 3 day

Start:
04/19/2024

Resort:
Bahamas Resort, 4 star

Per Person:
1350.00

Image Name:
reef2.jpg

Description:
Fake Trip added for testing purposes. Testing test

Delete Trip

Figure 6 - Deleting the custom trip.

Add Trip

Gale Reef



Emerald Bay, 3 stars

4 nights / 5 days only \$799.00 per person

Gale Reef Sed et augue lorem. In sit amet placerat arcu. Mauris volutpat ipsum ac justo mollis vel vestibulum orci gravida. Vestibulum sit amet porttitor odio. Nulla facilisi. Fusce at pretium felis.

Edit Trip

Delete Trip

Dawson's Reef



Blue Lagoon, 4 stars

4 nights / 5 days only \$1,199.00 per person

Dawson's Reef Integer magna leo, posuere et dignissim vitae, porttitor at odio. Pellentesque a metus nec magna placerat volutpat. Nunc nisi mi, elementum sit amet aliquet quis, tristique quis nisl. Curabitur odio lacus, blandit ut hendrerit.

Edit Trip

Delete Trip

Claire's Reef



Coral Sands, 5 stars

4 nights / 5 days only \$1,999.00 per person

Claire's Reef Donec sed felis risus. Nulla facilisi. Donec a orci tellus, et auctor odio. Fusce ac orci nibh, quis semper arcu. Cras orci neque, euismod et accumsan ac, sagittis molestie lorem. Proin odio sapien, elementum at tempor non.

Edit Trip

Delete Trip

Mega Reef



Barrier Island Resort, 5 stars

6 nights / 7 days only \$3,499.00 per person

The great barrier reef awaits! Cras orci neque, euismod et accumsan ac, sagittis molestie lorem. Proin odio sapien, elementum at tempor non. Pellentesque a metus nec magna placerat volutpat.

Edit Trip

Delete Trip

Figure 7 - Custom trip successfully deleted.

The Angular and Express frameworks offer distinct structures and functionalities suited to different aspects of web development. The Angular application is built around a Single Page Application (SPA) model, utilizing components that encapsulate specific logic. This promotes reusability and adheres to the Separation of Concerns (SoC) principle. The SPA facilitates dynamic content updates via API calls without requiring full page reloads, creating a more responsive user interface.

Conversely, the Express framework employs a Model-View-Controller (MVC) pattern ideal for server-side rendering. This supports a multi-page experience, where each navigation triggers a server request to load specific content. While this can enhance content-specific rendering speed, it does not match the efficiency of SPA's non-reloading behavior.

The Angular SPA enhances user interaction by allowing for dynamic changes without page reloads, significantly improving user experience. It maintains state across interactions, optimizing Create, Read, Update, and Delete (CRUD) operations. Alternatively, Express benefits from improved Search Engine Optimization (SEO) due to better indexing and the ability for users to bookmark specific pages. However, its reliance on page reloads can slow navigation.

Testing involved thorough verification of all API endpoints using Postman and MongoDB Compass to ensure correct behavior and data integrity. Following backend validation, the application was tested in a live environment, confirming all workflows through server logging and network diagnostics for real-time updates and data integrity checks.