Texas Animal Rentals

Creator: Chase Theriot

**Application:**

I chose to create my application with the theme of animals rentals. The application is a mock application and I decided to do a business that was crazy yet a bit unique. The application hierarchy has a client and server folder. Both the client and the server side of the project have their own dependencies and necessary parts to the overall project. The application has the ability to Create, Read, Update, and Delete. The Application can Create a new rental request, Read current rentals (more that once) and the animals that are available, Update a rental request, and Delete a rental request.

**Client Dependencies:**

The client side of the application uses the following dependencies: axios (sending info to the backend), bootswatch (free bootstrap styles), vue, vue-router, vuex (state management). There are also a few dev-dependencies that I utilized. The dev-dependencies are all dependencies that are included with the webpack.

**Server Dependencies:**

The server side of the application utilizes dependencies that are mostly picked by myself and the way that I chose to build the project. The dependencies for the application: body-parser (to work with json data), cors, express, joi, jsonwebtoken (token that represents the user in vuex), mongodb, mongoose (for interacting with the database), and morgan. The only dev-dependency that is in use on the server side is nodemon, which restarts the server after saving a file. This is very handy when you are making a lot of changes.

**Components**

Active Rentals: The active rentals component pulls up the rentals that the current user has requested.

Animals: The animals component pulls all of the animals in the database along with their information.

AppNav: This is the navigation bar that is available when the user is logged in.

ContactUs: This component is a contact form that allows the user to contact the company. This however is not functional at the time of the demo. It would not be hard to implement if the application was deployed.

Help: This component is to help the user navigate the application and utilize the features of it.

Landing: This component is the landing page of the application and allows the user to pick the login or registration option.

Login: This component allows the user to login.

Register: This component allows the user to create a user and then redirects to the login.  
RentalCan: This component allows the user to cancel their rental.

RentalReq: This component allows the user to create a rental request.

RentalUpdate: This component allows the user to make changes and update their rental request.