Minh Tran – 6355049
COSC212 – Assignment 2

Dog Rental Service Web Application Report

Project Link: https://altitude.otago.ac.nz/mtrn/dog-rental-service

Dog Rental Service is the online dog rental shop that allows customers to make booking for available dogs on a selected date and time. In this assignment, we built onto the first assignment and added back-end server-side functions to the project such as processing of the booking details and administrator's activities including removing bookings, modifying the available dogs. The application used php in interactions with javascript and css, and the data are stored into json files.

To access the web pages, users can navigate to the webpage by the link: https://dev212.otago.ac.nz:8443/~mtrn/assignments/index.php. This page, however, is password protected when it is hosted form the university server. In addition, this webpage is different from the one provided in the requirement as the index.php file is in "assignments" folder, not in "assignment". This change is due to the provided available file structure on the server.

The most important function of this development is the data storage of the new booking. Previously, the bookings that customer made are stored in the LocalStorage of the web browser. This storage is removable and is not recorded into the source file that we used to store the database; therefore, the new booking was not displayed on the admin view. In this update, once the users have submitted their booking, their form will be processed with a server-side input validation check. Server-side input validation make sure the input is in the right format to be saved in the database and prevent injection attacks to the database storage. When the check are passed, the request is sent to write into the database, which is currently the json file that contains all the bookings. Since this json file is used to load page elements, the new booking is displayed in the admin page. This assignment requirements do not focus construction of database, therefore the required storage in json file is sufficient.

During the development, we have added the authentication for a unified account of administrator. The administrators can login using username and password as "admin" and "admin" respectively. The usernames and passwords are stored in the mysql database in the server. The Mysql database is more secured than the json file that are used to currently storing the bookings and animals. Currently, there is only one account above stored in the database to demonstrate the administration authentication. Once the users have logged in as administrator, they can have a view at the admin webpages.

We have reconfigured the admin web page into two pages called Admin Booking and Admin Dog. These webpages are only visible after the users have logged in. Admin Booking displays the existing bookings, including the bookings that are recently created from using the website. This page is where the administrator can confirm that new bookings has been recorded. The administrator is allowed to cancel the booking by clicking on the corresponding button in the table. On the second page, Admin Dog, the administrator can view the data of all the existing dogs, and they can add, edit or remove dogs. Currently, the process of adding and editing dogs are started and distinguished through disabling and enabling form functions. Similar to adding the new bookings function above, the management of booking and dogs are operated on the json files and are then displayed on the admin view as we go.

In addition, we have also added the temporary memory of input using php session. If the input does not pass the server-side check, the wrong input will be kept in session and loaded back into input area to aid users in fixing their error.

Minh Tran – 6355049
COSC212 – Assignment 2

Third-party code and libraries:

For this development of Dog Rental Service, we only used the server-side check functions provided in the lab files to validate the input. We still use the same third-party code and libraries form previous development, which are Leaflet, jQuery and jQueryUI.

Test Case:

Authentication:

- Testing wrong input will give out error message
- Testing right input will display logout form
- Admin pages are displayed in navigation bar after logged in
- Redirect to index page after logout if currently on admin pages
- Reload the page to the current one after logout if not on admin pages

Recording the booking

- Alert the errors if validation failed
- Alert the errors if cannot output to file
- Output to files new bookings

Adding the Dog

- Enabling the Dog Details input area and clear up the dog information area
- Dog displayed on view and on output file upon successful adding

Removing the Dog

- Displayed view dogs does not have the dog on success removal
- The output file does not have the removed dog

Editing the Dogs

- Enabling the Dog Details input area and load up the dog information when the corresponding editing button is pressed
- Change the dog ID input area to read-only
- Display dog on view and on output file upon successful editing

Cancelling a booking

- Displayed view booking does not have booking on success removal
- The output file does not have the removed booking

Ongoing issues:

- The sessions are maintained even as the page reload, which are as expected to fit its purpose of keeping the previously inputted data but can be better fine-tuned.
- There is no session to record the previously stored selected items while making a booking. This is due to the method to post the items are different than the other inputs.
- There is no set size for the viewing of booking and dog in admin view. Event though it is working alright with the current file size, it can grow extremely long in large database and render the web page difficult to use.
- Server-side validation of the data type date and time are currently being ignored.