Lee Paolucci, Alex Tuttle, Mitchell Ruple

Term Project

Database Management

Dr. Cheng

Final Report- Zippys Room Search

Implementation for Zippy’s Rooms includes a front end of HTML and CSS plus the server scripting language of PHP and the back end of MySQL and Python. Our web page allows users to reserve a room, delete a room, update a reservation, and display available rooms. Our advance functions we included in our project is the search function which searches through the grabbed data from myAkron’s website and displays to the users all the classroom times that are booked. Another advance function we included is the login interface for the user, imitating if a user were to put in their Akron login. Our Python package Selenium inputs the time into the myAkron’s class search page and runs twice for Monday, Wednesday, Friday and then Tuesday, Thursday. The search results are then inputted into our database for the user to access those times.

Within the website, users are greeted with a log in page. No matter what page the users attempts to reach, they are redirected to this page. Users can log in with their student id number and password. This data would be implemented by the University, so a sign-up option is not available to prevent random people from accessing the website. Once the user has logged in, they can choose from a variety of features on the left side of the page.

The first option is the advanced search function. This is an advanced function because it deals with team constraints and specifying a date. In the database, there is a separate column for each day. Then, the column contains short integers, which only allow 0 or 1, signifying if that day is reserved. This helped make it easier for inserting data into the database; however, it was more difficult to search for a particular day.

The next feature allows users to view their current reservations. Since the user is logged into the system, they do not need to enter any new information on this page. Direct access to this page allows users to view their current reservations. This is implemented by taking the id of the user and inputting it into a search query in the user input table. Then the results of this query are output to the page on load.

The third feature allows users to reserve a room. When a user chooses this option, they are greeted with a form that includes their student ID, which cannot be changed, a start time, end time, room ID, and a drop-down option for the day of the week. Then this form is converted into a SQL query that is inserted into the user table. However, before the data is inserted, two more queries are running which check the reservation does not overlap a class time or another reservation. This is accomplished by check start and end time, along with the date and classroom.

The next feature allows users to update their current reservation. This was quite difficult to implement, because it required pulling a current reservation from the table, then checking that the new requirements did not overlap a reservation that already exists, and then updating the reservation with the new information. This works by holding the old and new details of the reservation and comparing each detail to make sure the reservation can be updated. However, this requires a lot of code because each date must go through an if statement to compare the date to the Boolean values in the table. Thus, this became the longest file because of the double checking of dates.

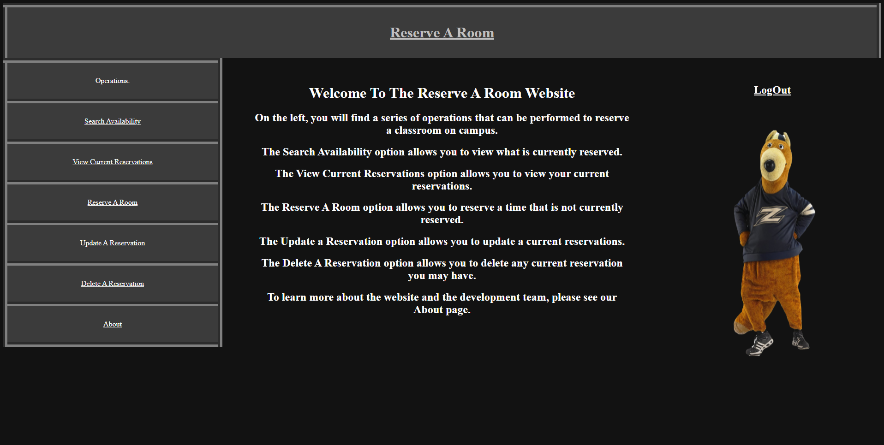
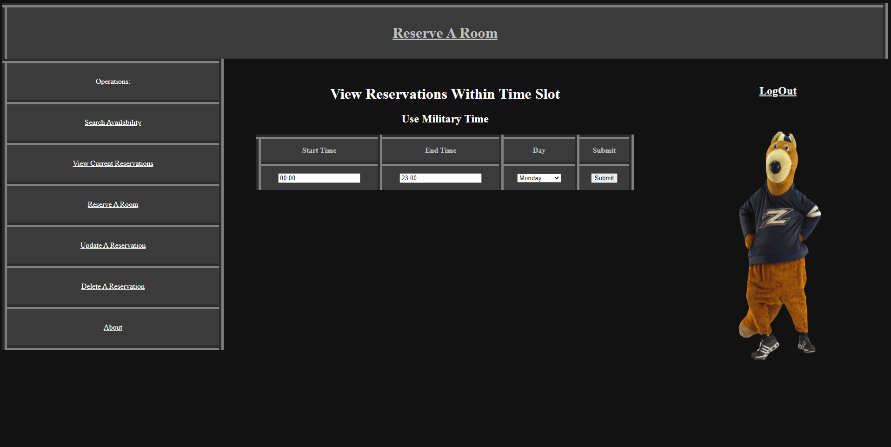
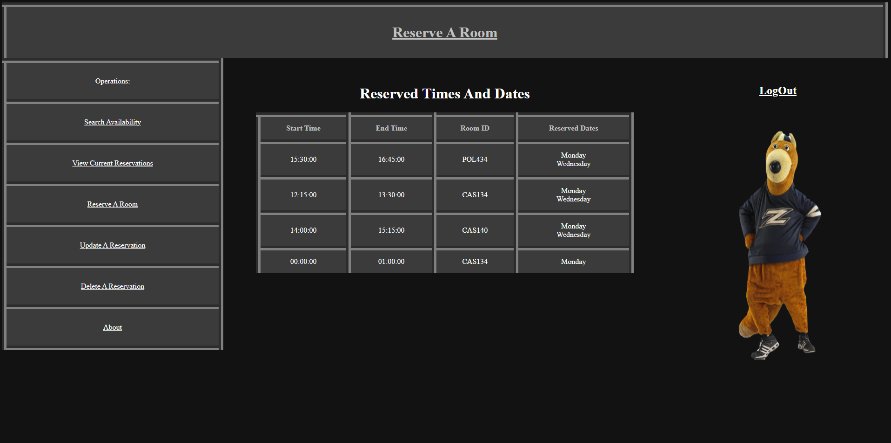
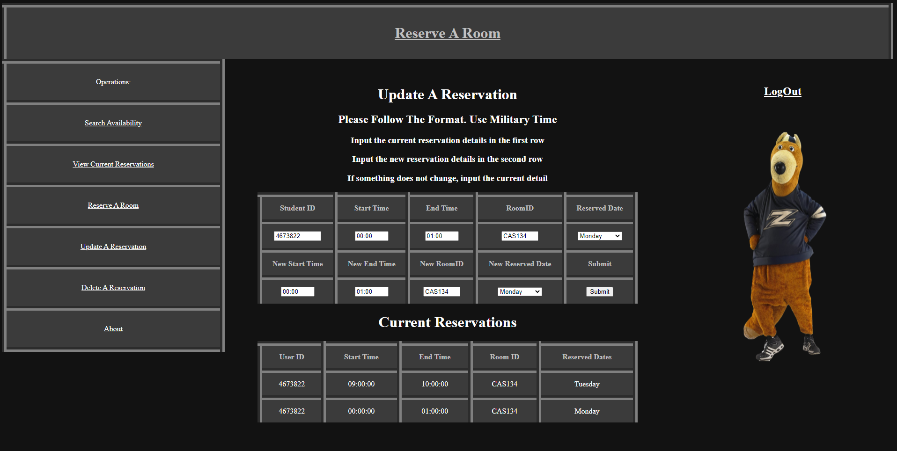
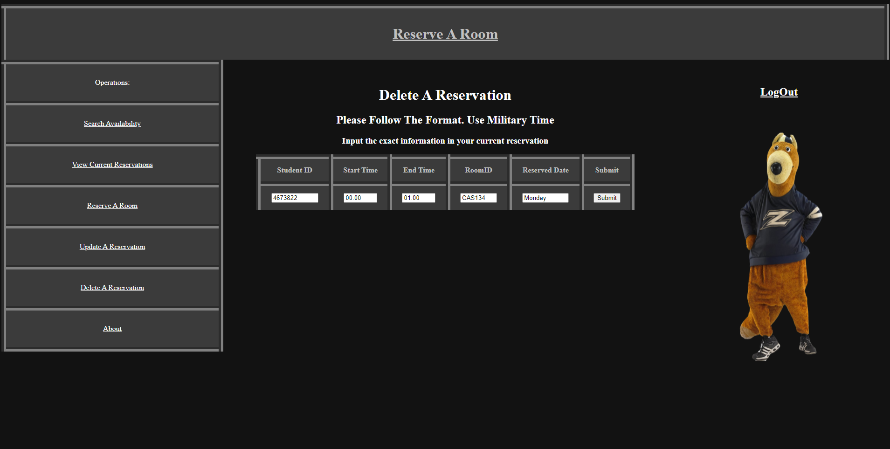
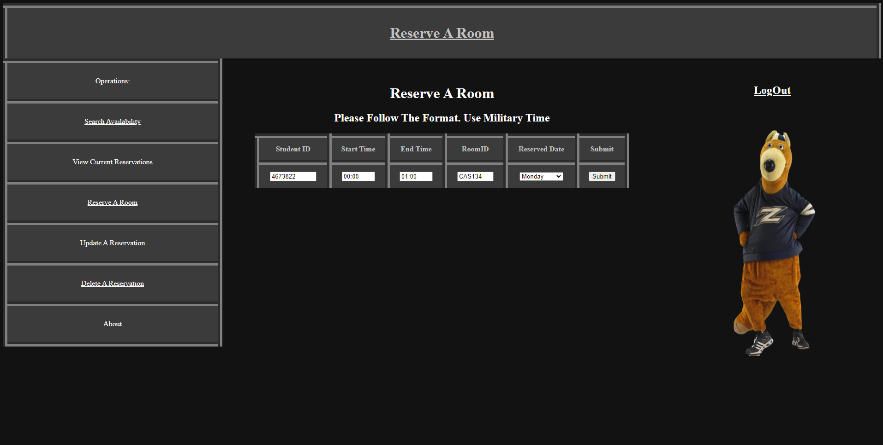
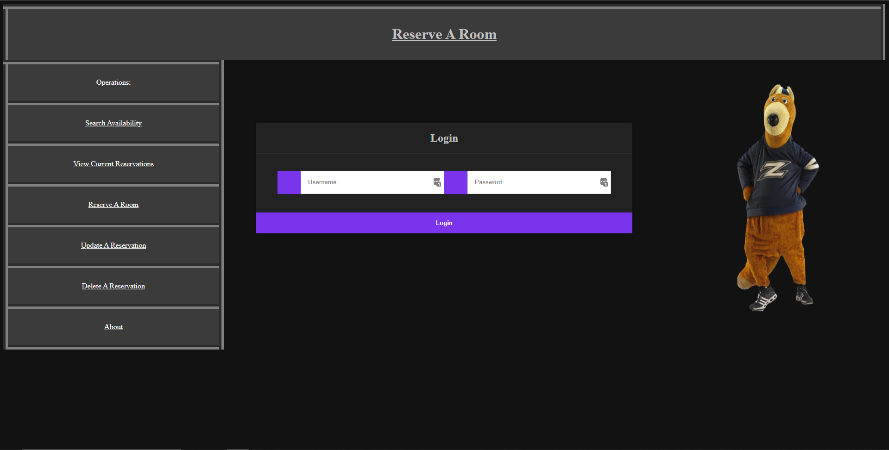
A final feature allows users to delete a current reservation. The user is greeted with a similar form to the add a reservation page, where they must enter each value into the table to delete their reservation. Once the user enters the correct data, the reservation is removed from the table in the database. Then, the user is greeted with a page that shows their updated current reservations.

Finally, the user can view the about page to read a simple explanation of each available feature. The heading at the top of each page also redirects the user to the about page. Once a user has completed everything they want to do, they can log out of the website by clicking the log out button on the right side of every page.

The contributions done to the project was Lee handled all the python package and Alex/Mitch on the rest. Alex handled mostly the front end with web development and setting up the backend database, while Mitch was learning because he has never experienced any other language except C++ and Alex had prior exposure to HTML and CSS.

A few lessoned we learned included learning new languages such as PHP. While Alex had a little experience with PHP in the past, he learned a lot more about the language and became very confident in understanding how it works. Lee also learned a lot about Python during the project, having little knowledge of the language before this. He, as well, became more confident in his ability to write in Python. Overall, we learned how to work together and complete a project as a group.

Some possible future work would be to implement a better user interface. Currently, users must enter information for each feature. However, in the future this could change to a table format, where users can directly insert, update, and delete from the table. Also, weekly updates that delete the reservations from the previous week would be a good feature to implement. However, due to time constraints, we were not able to implement these features, but would make the web application more useful for school and other companies.

Screenshots