Project Descriptions

* Version Controlled Songwriting App
  + During the songwriting process, songwriters frequently find themselves with an abundance of song fragments, chord progressions, and other theoretical or aural musical inspirations with no effective means of documenting and organizing these elements. **Tracking the progression of a song and documenting how it changes over time is a time-consuming process, requiring writers to continually organize their ever-expanding library of files. Often the user is forced to delete files containing plausible song iterations, or transfer their ideas to a different device, making the comparison of different versions of a given song difficult.** Our solution to this problem is a system that provides version control for songwriters. Our system would use a tree structure to record the development of the project, providing songwriters with easy access to their audio and text files. Additionally, this system would allow users to examine previous versions of their project and track the development of their work. Key features of this system would allow users to contain all the information of the song at that point in time (lyrics, audio recording, photos/videos of song being played, keyboard settings, etc). As well, this system would provide an audio recording interface that allows the user to have lyrics on their phone screen as they are recording their song iterations.
* React Motel Management CRUD App
  + A dockerized web app using React and MUI for the frontend, Node/Express for the backend, MySQL for the database, and Auth0 for authorization and account creation. This app was the term project of my software engineering class and was made to manage operations for the Twilite Motel, located in beautiful Prince Albert, Saskatchewan. Functionalities include manager, employee, and customer account creation with varying levels of access according to account type, reserving rooms both online as a customer and from the front desk as an employee, and employee shift scheduling.
* LeNet CNN for Plant Image Classification
  + A LeNet deep learning model created to classify images of sorghum plants from the Sorghum-100 dataset as a baseline of performance in comparison with more sophisticated CNNs. The model was created using Keras API for TensorFlow, and implements data augmentation using flips, rotations, and zoom. The Sorghum-100 dataset is comprised of 100 different classes, where each class label is the ID of 1 of the 100 genetically distinct cultivars, and was pre-processed with CLAHE (contrast-limited adaptive histogram equalization) to improve contrast within each image and alleviate the obstruction of plant features from the effects of ambient lighting conditions.