

# Lessons learned

## Order of operations

Its beneficial to have the base functionality of multiple parts of the software created instead of attempting to fully finish one at a time

## Testing

Frequent testing after every change helps for a smoother merging process

## Divide and Conquer

Dividing work based on team members skill sets makes for more efficient development