Report of the Final Project of Database Class

- Language and Frameworks:
    - The main program is written in python with flask framework and pymysql library for interacting with the database. For hash encoding, I used the werkzeug security model in the template. For the database, I choose MySQL for simplicity. For the front-end display, I used html, css and javascript.
- Schema:
    - The schema I used for the project is the schema provided on the brightspace, with primary key and foreign constraint as states. For acceptDonation function, I added a trigger in order to make pieceNum increase automatically from 1 with one itemID if there are pieces for the item. If not, the pieceNum will be automatically assigned 0.
- Queries:
    - findItem:
        - SELECT * FROM item NATURAL JOIN piece WHERE item.itemID = %s
    - findOrder:
        - SELECT * FROM item JOIN itemin ON item.itemID = itemin.itemID LEFT JOIN piece ON item.itemID = piece.itemID WHERE orderID = %s
    - orderHistory:
        - SELECT * FROM ordered WHERE supervisor = %s OR client = %s
    - acceptDonation:
        - SELECT roleID FROM act WHERE userName = %s

        - SELECT DISTINCT mainCategory, subCategory FROM category
        - SELECT roleID FROM act WHERE userName = %s
        - INSERT INTO item (iDescription, color, isNew, hasPieces, material, mainCategory, subCategory) VALUES (%s, %s, %s,%s, %s, %s,%s)
        - INSERT INTO donatedBy (itemID, userName, donateDate) VALUES (%s, %s, %s)
        - INSERT INTO piece (itemID, roomNum, shelfNum) VALUES (%s, %s, %s)
    - phoneNumbers:
        - INSERT INTO personphone(userName, phone) VALUES (%s, %s)
        - SELECT phone FROM personphone WHERE userName = %s
    - addToCurrentOrder:
        - SELECT DISTINCT mainCategory, subCategory FROM category
        - SELECT COUNT(*) FROM ordered WHERE orderID = %s
        - SELECT itemID, iDescription, color, material FROM item WHERE mainCategory = %s AND subCategory = %s AND itemID NOT IN (SELECT itemID FROM itemin)
        - INSERT INTO itemin (orderID, itemID) VALUES (%s, %s)
        - SELECT item.itemID, iDescription, color, material FROM item JOIN itemin ON item.itemID = itemin.itemID WHERE orderID = %s
    - loadUser:
        - SELECT * FROM person WHERE cid = %s
    - register:

        - SELECT 1 FROM person WHERE userName = %s
        - INSERT INTO person (fname, lname, userName, password, email) " "VALUES (%s, %s, %s, %s, %s)
        - INSERT INTO act (userName, roleID) " "VALUES (%s, (SELECT roleID FROM role WHERE rDescription = %s))
    - login:
        - SELECT * FROM person WHERE userName = %s
- Difficulties and Lessons:
    - The trigger can be difficult to deal with since there are multiple situations where items can have multiple or no pieces and the assignment of pieceNum should be dealt with carefully.
    - The javascript part can also be challenging, for example, when chosen a main category in the toggle bar, the sub category selections must be under the main category, which requires a script to achieve the functionality.