

Languages and frameworks you used: python flask pymysql and html taught in this class

Changes I made to the schema, and their purpose:

I deleted the 'shelf' in 'Location' as there is already a 'shelfNum' and 'shelfDescription', we can set shelfNum=0 to handle the situation without a specific shelf assigned.

Additional constraints, triggers, stored procedures, etc:

```
`photo` blob,  
`isNew` tinyint(1) DEFAULT '1',  
`hasPieces` tinyint(1) DEFAULT NULL
```

I changed the type of 'photo' from Varchar to blob, and the type of 'isNew' and 'hasPieces' are automatically altered from Boolean to tinyint.

I also modify the stored procedure of password with salt and hash:

```
hashed_password = bcrypt.hashpw(password.encode('utf-8'), bcrypt.gensalt())
```

The main queries for each of the features I implemented:

1. Login & User Session Handling:

/loginAuth

```
SELECT password FROM Person WHERE userName = %s
```

/registerAuth

```
SELECT * FROM Person WHERE userName = %s
```

```
INSERT INTO Person (userName, password, fname, lname, email)
```

```
VALUES (%s, %s, %s, %s, %s)
```

```
INSERT INTO Act (userName, roleID)
```

```
VALUES (%s, %s)
```

2. Find Single Item:

```
SELECT Item.mainCategory, Item.subCategory, Item.iDescription AS itemDescription,  
Piece.pieceNum, Piece.roomNum, Piece.shelfNum, Piece.pNotes, Piece.pDescription AS  
pieceDescription, Piece.length, Piece.width, Piece.height FROM Piece NATURAL JOIN Item  
WHERE Piece.ItemID = %s;
```

3. Find Order Items:

```
SELECT Item.ItemID, Item.mainCategory, Item.subCategory, Item.iDescription AS  
itemDescription, Piece.pieceNum, Piece.pDescription AS pieceDescription, Piece.length,  
Piece.width, Piece.height, Piece.roomNum, Piece.shelfNum, Piece.pNotes FROM Piece  
NATURAL JOIN Item NATURAL JOIN ItemIn WHERE ItemIn.orderID = %s ORDER BY  
ItemID, pieceNum;
```

4. Accept Donation:

```
SELECT userName FROM `Act` WHERE userName = %s AND roleID=2
```

```
SELECT userName FROM `Act` WHERE userName = %s AND roleID=1
```

```
SELECT * FROM Category WHERE mainCategory = %s AND subCategory = %s
```

```
INSERT INTO Category (mainCategory, subCategory)
```

```
VALUES (%s, %s)
```

```
INSERT INTO Item (iDescription, photo, color, isNew, hasPieces, material, mainCategory, subCategory)
```

```
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)
```

```
INSERT INTO DonatedBy (ItemID, userName, donateDate)
```

```
VALUES (%s, %s, CURDATE())
```

/enterPiece

```
SELECT * FROM Location WHERE roomNum = %s AND shelfNum = %s
```

```
INSERT INTO Location (roomNum, shelfNum)
```

```
VALUES (%s, %s)
```

```
INSERT INTO Piece (ItemID, pieceNum, pDescription, length, width, height, roomNum, shelfNum, pNotes)
```

```
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)
```

5. Start an order

```
SELECT userName FROM `Act` WHERE userName = %s AND roleID = 2
```

```
SELECT userName FROM `Act` WHERE userName = %s AND roleID = 4
```

```
INSERT INTO Ordered (orderDate, orderNotes, supervisor, client)
```

```
VALUES (CURRENT_DATE, %s, %s, %s)
```

6. Add to current order (shopping)

```
SELECT DISTINCT mainCategory FROM Item
```

```
SELECT DISTINCT subCategory FROM Item WHERE mainCategory = %s
```

```
SELECT ItemID, iDescription FROM Item WHERE mainCategory = %s AND subCategory = %s AND ItemID NOT IN (SELECT ItemID FROM ItemIn)
```

```
INSERT INTO ItemIn (orderID, ItemID)
```

```
VALUES (%s, %s)
```

Difficulties I encountered, lessons learned:

When I selected main category and sub-category in shopping, went to the shopping website and pressed the button to submit it, the main category and sub-category disappeared. To avoid this situation, I created session variable main category and sub-category to keep them in the session. In this way, when I select an item to add them to an order, the main category and subcategory are kept. Same with the orderID, I also created a session variable for it to avoid the autoincrement for it every time we refreshed the website or went back to that page. I also learned that I should remember to pop up the session variables accordingly to avoid always adding item to the same orderID or for the same client and to add a None after the variable popped up like `session.pop('orderID', None)` to avoid error.

Which team members did what:

All done by Vivian Zhao (myself), I don't have a team member.