Languages and Frameworks Used

- **Languages**: Python, SQL
- **Frameworks**: Flask (for web application), Flask-Login (for user authentication), MySQL (for database management)

Changes Made to the Schema

- Mostly using existing schema
    - hasPieces in the Item table to store the number of pieces an item has.

Additional Constraints, Triggers, Stored Procedures

- No stored procedures or triggers were created for this implementation. All logic is handled within the Flask application using prepared SQL queries to prevent SQL injection.

Main Queries for Features Implemented

- Login user:

```
'SELECT * FROM Person WHERE userName = %s', (username,))
```

- Register:

```
'SELECT 1 FROM Person WHERE userName = %s", (username,))

"INSERT INTO Person (userName, password, fname, lname, email) "
"VALUES (%s, %s, %s, %s, %s)",
```

- Find item:

```
 "SELECT roomNum, shelfNum FROM Piece WHERE ItemID = %s", (item_id,)
```

- Donation:

```
'SELECT 1 FROM Act WHERE userName = %s AND roleID = 'staff'",

'SELECT * FROM Person WHERE userName = %s', (donor_id,))

"INSERT INTO Item (iDescription, color, material, mainCategory, subCategory, hasPieces) "
"VALUES (%s, %s, %s, %s, %s, %s)",

"INSERT INTO Piece (ItemID, pieceNum, pDescription, length, width, height, roomNum, shelfNum, pNotes) "
"VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s)",

"INSERT INTO DonatedBy (ItemID, userName, donateDate) VALUES (%s, %s, CURDATE())",
(item_id, donor_id)
```

- Find order:

```
SELECT i.ItemID, i.iDescription, p.roomNum, p.shelfNum
FROM Item i
JOIN ItemIn ii ON i.ItemID = ii.ItemID
JOIN Piece p ON i.ItemID = p.ItemID
WHERE ii.orderID = %s
```

- Start an order:

```
"INSERT INTO Ordered (orderDate, supervisor, client) VALUES (CURDATE(), %s, %s)",
```

Difficulties Encountered and Lessons Learned

- Connecting to the database
- Modifying templates for the right display/user interface
- Debug

Team Member Contributions

- All features implemented individually