# Project Report

TPG team: Jeffery Zeng, Jiaxin Sun, Yutong Tian,

December 9, 2024

## Introduction

The project aims to build a web interface for Welcome Home, a non-profit organization that assists refugees and asylum seekers in tracking donations, orders, deliveries, clients, donors, and volunteers.

## 1 Languages and Frameworks Used

- TypeScript
- Python
- React(front-end)
- Flask(back-end)

## 2 Schema Changes

- Changed the password attribute of the Person table from varchar(100) to varchar(512) to accommodate the storage of hashed passwords.
- Add an index idx_item_category to Item table to speed up the search.

## 3 Additional Constraints

- Triggers
  - Add triggers to trace the change of hasPieces in Item table
  - Add triggers to restrict the selectable options for the status attribute in the Delivered table.
  - Add triggers to trace the change of hasPieces in Item table

## 4 Main Queries

For each feature implemented, present the main SQL queries:

- Feature 1: Login & User Session Handling

```
-- SQL query to check if the username already exists
SELECT *
FROM Person
WHERE userName = %s;
```

```
-- SQL query to insert a new user into the Person table
INSERT INTO Person (userName, password, fname, lname, email)
VALUES (%s, %s, %s, %s, %s);
```

```
-- SQL query to insert phone numbers into the PersonPhone table
INSERT INTO PersonPhone (userName, phone)
VALUES (%s, %s);
```

```
-- SQL query to insert user role into the Act table
INSERT INTO Act (userName, roleID)
VALUES (%s, %s);
```

```
-- SQL query to fetch user information for login
SELECT *
FROM Person
WHERE userName = %s;
```

```
-- SQL query to fetch the role of the user
SELECT roleID
FROM Act
WHERE userName = %s;
```

- Feature 2: Find Single Item

```
-- SQL query to find item locations excluding pieces ready for delivery
SELECT p.pieceNum, p.pDescription, p.length, p.width, p.height,
p.roomNum, p.shelfNum, p.pNotes
FROM Piece p
WHERE p.ItemID = %s AND NOT (p.roomNum = -1 AND p.shelfNum = -1);
```

- Feature 3: Find Order Items

```
-- SQL query to find items in the specified order
-- with additional piece information
SELECT
    i.ItemID,
    i.iDescription,
    p.pieceNum,
    p.pDescription,
    p.length,
    p.width,
    p.height,
    p.roomNum,
    p.shelfNum
FROM Item i
NATURAL JOIN ItemIn ii
NATURAL JOIN Ordered o
NATURAL JOIN Piece p
WHERE o.orderID = %s;
```

- Feature 4: Accept Donation

```
-- SQL query to check if the user is a registered donor
SELECT * FROM Act WHERE userName = %s AND roleID = 'donor';

-- SQL query to check if the user is a registered donor
SELECT * FROM Act WHERE userName = %s AND roleID = 'donor';

-- SQL query to insert a new item record
INSERT INTO Item (iDescription, color, isNew, hasPieces,
material, mainCategory, subCategory)
VALUES (%s, %s, %s, %s, %s, %s, %s);

-- SQL query to insert a piece associated with the item
INSERT INTO Piece (ItemID, pieceNum, pDescription, length,
width, height, roomNum, shelfNum, pNotes)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s);

-- SQL query to insert the donation record into DonatedBy
INSERT INTO DonatedBy (ItemID, userName, donateDate)
VALUES (%s, %s, %s);
```

- Feature 5: Start an Order

```
-- SQL query to create a new order
INSERT INTO Orders (clientUsername, orderID)
VALUES ('input_clientUsername', 'new_orderID');
```

- Feature 6: Add to Current Order (Shopping)

```
-- SQL query to add an item to the current order
INSERT INTO OrderItems (orderID, itemID)
VALUES ('current_orderID', 'input_itemID');
```

- Feature 7: Prepare Order

```
-- SQL query to search orders by order number
SELECT * FROM Ordered WHERE orderID = %s;

-- SQL query to search orders by client username
SELECT * FROM Ordered WHERE client = %s;

-- SQL query to update the location of all pieces
-- associated with the order to (-1, -1)
UPDATE Piece
SET roomNum = -1, shelfNum = -1
WHERE ItemID IN (
    SELECT ItemID FROM ItemIn WHERE orderID = %s
);
```

- Feature 8: User's Tasks

```
-- SQL query to retrieve orders
-- where the current user is a client
SELECT o.orderID, o.orderDate, o.orderNotes, o.supervisor,
d.username AS deliverer, d.status, d.date
FROM Ordered o
LEFT JOIN Delivered d ON d.orderID = o.orderID
WHERE o.client = %s;

-- SQL query to retrieve orders where the
-- current user is a supervisor
SELECT o.orderID, o.orderDate, o.orderNotes, o.supervisor,
d.username AS deliverer, d.status, d.date
FROM Ordered o
LEFT JOIN Delivered d ON d.orderID = o.orderID
WHERE o.supervisor = %s;

-- SQL query to retrieve delivered orders with
-- order date and notes
SELECT o.orderID, o.orderDate, o.orderNotes,
o.supervisor, d.username AS deliverer, d.status, d.date
FROM Delivered d
NATURAL JOIN Ordered o
WHERE d.userName = %s;
```

- Feature 9: Rank System

```
-- Defining the task of volunteer is delivery
-- SQL query for ranking volunteers‘  deliveries in a specific time period
SELECT userName, COUNT(*) AS con
FROM delivered NATURAL JOIN act
Where roleID = 'volunteer' AND date >= %s AND date <= %s
GROUP BY userName
Order BY con DESC
```

- Feature 10: Update Enabled

```
-- SQL query to update order status
UPDATE delivered
SET status = %s, date = %s
WHERE orderID = %s AND userName = %s
```

## 5 Difficulties Encountered

- One of the first difficulties we encountered was a cookie loss issue related to Cross-Origin Resource Sharing (CORS). This problem arose when our frontend and backend servers were hosted on different domains.
- To resolve this issue, we added the necessary request headers in our frontend server during requests. Additionally, we enabled CORS in our Flask backend by installing the ‘flask_cors‘ package and configuring it appropriately to allow requests from our frontend origin.
- This experience taught us the importance of understanding CORS and its implications on cookie handling and session management when dealing with separate frontend and backend services.
- We also learned to thoroughly test interactions between the frontend and backend in a cross-origin context to preemptively identify and address similar issues.

## 6 Team Member Contributions

- Jeffery Zeng: Feature 7 & 8
- Jiaxin Sun: Feature 5 & 6
- Yutong Tian: Feature 9 & 10