Project Report

Group Members:

- **Meet Oswal** (mo2532)
- **Nikhil Soni** (ns6062)
- **Utkarsh Jain** (uj299)

1. Languages and Frameworks Used

- **Frontend**: React.js, HTML, CSS, JavaScript
- **Backend**: Python (Flask), SQL
- **Database**: MySQL

2. Changes Made to the Schema

- **Additions**:
  No Change in the Schema Made

3. Additional Constraints, Triggers, and Stored Procedures

- **Constraints**:
  - Implemented foreign key constraints to ensure data consistency.
- **Triggers**:
  - Created trigger before insertion into person table to check whether new email or phone already exists in the database, and if yes throw custom error.
- **Stored Procedures**:
  - Created a stored procedure to update order status given the updater's role is Staff / Volunteer and they are related to this order.

4. Main Queries for Each Feature

- **Login and User Session Handling**:
  SELECT * FROM person WHERE username = 'meetoswal' AND password = 'password
- **Find Single Item**:
  SELECT * FROM Items WHERE itemID = 1;

- **Find Order Items**:
  SELECT * FROM orders WHERE orderID = 5;
- **Accept Donation**:
  INSERT INTO Item (iDescription, photo, color, isNew, hasPieces, material, mainCategory, subCategory) VALUES (“Item Description”, <blob image>, “Red”, True, False, “Steel”, “Furniture”, “Table”);(Help Meet)

  INSERT INTO DonatedBy (ItemID, userName, donateDate) VALUES (1, “meetoswal”, “2024-12-08”)

  INSERT INTO Piece (ItemID, pieceNum, pDescription, length, width, height, roomNum, shelfNum, pNotes) VALUES (1, 1, “Table”, 10, 10, 10, 1, 1, “Notes”)

- **Start an Order:**

  SELECT *
  FROM act
  WHERE userName = %s and roleID = 'Client'

- **Add to current order:**

  SELECT *
  FROM item natural left join itemin
  WHERE orderID is null and ((mainCategory = 'Books' and subCategory in ('Comedy')) or (mainCategory = 'Clothing' and subCategory in ('Men'))) order by ItemID desc limit 11 offset 10

- **Prepare Order:**

  INSERT INTO ordered (orderDate, orderNotes, supervisor, client) VALUES (‘2024-12-08’, “Notes, “meetoswal”, “nikhilsoni”)
  INSERT INTO itemin (ItemID, orderID, found) VALUES (1, 1, 1)
  INSERT INTO delivered (userName, orderID, status, date) VALUES (‘utkarshjain’, 1, 1, ‘2024-12-13’)

- **User Task:**
  with orderIDs as (SELECT distinct orderID
  FROM `ordered` natural join itemin natural join delivered
  WHERE client = 'meetoswal' or supervisor = 'meetoswal' or userName = 'meetoswal')
  SELECT orderID, ItemID, iDescription
  FROM orderIDs natural join (itemin natural join item)

ORDER BY orderDate DESC

- **Rank System:**
  SELECT userName, count(orderID) as 'count'
  FROM (select * from act where roleID = 'Volunteer') as volunteers natural left join (delivered natural join ordered)
  WHERE orderDate >= ‘2024-10-10’ and date <= ‘2024-12-12’
  GROUP BY userName
  ORDER BY count(orderID) desc
  LIMIT 10

- **Update enabled!:**
  call UpdateOrderStatus(1, ‘meetoswal’, ‘Delivered’)

5. Difficulties Encountered and Lessons Learned

- **Difficulties**:
  - Debugging session handling with Flask due to inconsistent session storage.
  - Encountered issues with resolving CORS (Cross-Origin Resource Sharing) errors during frontend-backend communication.
  - Managing React state for complex components like ranking and order handling.
  - SQL query optimization.
- **Lessons Learned**:
  - Improved understanding of REST API design and implementation.
  - Gained experience with React Context hook for state management.
  - Learned to use bcrypt for secure password hashing.
  - Enhanced understanding of Flask's modular design and its blueprint functionality for organizing and scaling applications

6. Team Member Contributions

- **Meet Oswal**:
- **Accept Donation**
- **Rank System**
- **Update enabled**

- **Nikhil Soni**:
- **Login & User Session Handling:**
- **Add to current order (shopping)**

- **User’s tasks**

- **Utkarsh Jain**:
- **Find Single Item**
- **Find Order Items**
- **Start an order**
- **Prepare order**