Instructions

Links for our APP's repos:
Backend: https://github.com/csc301-fall-2020/assignment-1-4-accelsnow-altair59-backend
WEB Frontend: https://github.com/csc301-fall-2020/assignment-1-4-accelsnow-altair59-web
MOB Frontend: https://github.com/csc301-fall-2020/assignment-1-4-accelsnow-altair59-mobile

Design Overview

- We decide to implement a checkout system backed up by an item inventory supported by MySQL database. Any user from WEB end or MOBILE end of our APP can search to add item, modify cart, and see the calculated total of the items (involving individual item discount, global cart discount, and global tax rate as parameter), and each session ends after the user confirms the receipt. However, considering that manager operations on the inventory (editing certain fields of item, removing item, adding item under restriction, and editing global taxes & discounts) are sophisticated to be performed and table with massive information are not visually clean and tidy on MOBILE end, we decide to implement access (of course with authorization) to manager portal only on WEB end.

Functionality Instruction

============================BOTH WEB & MOBILE APP==========================

- User can type in the correct item id (if the user knows the target item's id) or standard item name (e.g. coke, sprite…) to add that item (quantity 1) into the cart. If the item is already in the cart, increment that item's quantity by 1. Error messages will be given if such item does not exist or is currently out of stock.

- User can see proper information of all items in the cart.

- User can edit the quantity of each item in the cart manually in the range of [0, stock]. Item will be removed from the cart if invalid input is given.
  REQUIREMENT:
  0 <= quantity <= stock

- User can see the net total of all items in the cart in real time (update as user changes quantity).

- User can click the checkout button to see the receipt which includes more detailed information (cart discount, cart tax rate, total, and time of checkout). If the user clicks the confirm button, a purchase is completed and the cart will be cleaned and inventory will be updated correctly (thus entering next purchase), else user is sent back to the cart (the purchase remains pending).

- If someone else also bought this item before the user checks out, the quantity of that item in the cart will be adjusted if needed (e.g. if A wants to buy 2 cokes but the stock of coke is changed from 2 to 1 since B buys 1 coke before A checks out, then when checks out A will be notified of this change and coke quantity in A's cart will be updated to 1 accordingly).

===========================BELOW ARE WEB APP ONLY=========================

- In order to login as Manager, user need this Info: {username: manager, password: Passw0rd123}. User can click the manager portal to log in as manager, and be redirected to manager portal page, where user can click the log out button to quit manager session and be redirected to checkout page. All unauthorized accesses to manager page are prohibited.

- Manager can type in valid information about a new item manager comes up with and click the add button to add the new item to the inventory. If invalid values are given, the add button will be disabled. Other errors only checkable in database (e.g. duplicate key…) will be reported as an alert if occurred.
- REQUIREMENT:
- name: String
- price: Float && 0 <= price
- discount: Float && 0 <= discount <= 1
- stock: Integer && 0 <= stock

- Manager has all access to all items' information. In addition, manager can edit one item's price, discount, and stock in-line or remove this item from the inventory. Errors will be caught and reported as an alert.
- REQUIREMENT:
- price: Float && 0 <= price
- discount: Float && 0 <= discount <= 1
- stock: Integer && 0 <= stock

- Manager can edit the global tax rate or discount given valid input. Errors will be caught and reported as an alert.

- REQUIREMENT:
- discount: Float && 0 <= discount <= 1
- tax_rate: Float && 0 <= tax_rate

Frontend Component Design

===========================BOTH WEB & MOBILE APP=========================

- Checkout page:
  - An item search bar to add items into the cart
  - A table of items in the cart with proper information
  - A calculator of total cost of items in the cart

- A checkout button
- A modal displaying receipt information
- A log in button to enter manager session

==========================BELOW ARE WEB APP ONLY========================

- Manager page:
  - An form to add a new item
  - A table of all the items in inventory with proper information with in-line edit of item information and removal of item
  - A form to edit global discount and tax rate
  - A log out button to quit manager session

Backend Model Design

- The database models we implement:

```
Manager: {
    username: String, // username
    password_hash: String // passoword
}
Item: {
    id: Integer, // unique item id
    name: String, // item name
    discount: Float, // item discount rate should be in range [0,1]
    price: Float, // item price should be in range [0,
    stock: Integer // iten stock should be in range [0,
}
Checkout: {
    id: Integer, // unique checkout id
    tax_rate: Float, // global tax rate should be in range [0,
    discount: Float // global discount should be in range [0,1]
}
```

Testing Instruction

- Testing for backend logic

  All tests are written in Pytest, the testing directory is "assignment-1-4-accelsnow-altair59-backend/tests/", including:

  1. "conftest.py" for setting up testing environment.
  2. "test_db_integrity.py" for checking database integrity.
  3. "test_api_correctness.py" for checking api functionalities.
- Testing for App functionality
  1. Please refer to section Functionality Instruction for all operations and behaviours.