Test Plan for Peer Testing

Team Information

Group 5: Food Delivery Dashboard for Restaurants

Members:

- Enrique Perez (egperez@uwm.edu)
- George Wiedemann (wiedema9@uwm.edu)
- Israel Adigun (imadigun@uwm.edu)
- Kevin Santamaria (romero25@uwm.edu)

Preferred Communication: please reach out to us via a message in Microsoft Teams.

System Overview

Our project is a restaurant order management web application designed to streamline and simplify the delivery order process for restaurants that use third-party applications. The application consolidates incoming orders from multiple third-party food delivery platforms such as DoorDash, Uber Eats, Grubhub, and others into a single, unified dashboard.

Purpose:
This system is intended for restaurant staff who manage delivery orders through third-party applications. Instead of switching between different apps or tablets for each delivery service, they can view and manage all active pickup orders in one place. This assists in queueing orders alongside in-house orders, improving overall workflow efficiency while preventing longer waittimes.

Key Features:

- Centralized display of active orders from different platforms.
- Ability to mark orders as completed.
- Visual indicators to help staff prioritize tasks.
- Simple and intuitive user interface designed for fast-paced environments.

Access Instructions

Pre-requisites:

- VSCODE or other Development Environment.
- Python 3.13 or higher (as some installs are version specific)
    - Python Release Python 3.13.1 | Python.org

Getting the Project Started:

1. Clone project from: egperezuwm/FoodApp: FoodDeliveryApp
    - In terminal, "git clone egperezuwm/FoodApp: FoodDeliveryApp"
2. Create your Virtual Environment
    - In terminal, Navigate to backend "cd backend"
    - WINDOWS: In terminal, "python -m venv .venv"
    - MACOS: In terminal, "python3 -m venv .venv"
3. Start your Virtual Environment
    - WINDOWS: In terminal, ".venv/Scripts/activate"
    - MACOS: In terminal, ".venv/bin/activate"
4. Install requirements.txt
    - WINDOWS: In terminal, type: pip install –r "requirements.txt"
    - MACOS: In terminal, type: pip3 install –r "requirements.txt"
5. Start the backend.
    - In terminal, activate server by "python manage.py runserver"
    - IF YOU NEED A DATABASE, in terminal type:
        i. python manage.py makemigrations
        ii. Python manage.py migrate
6. Open ANOTHER Terminal to start frontend.
    - In terminal, Navigate to frontend"cd frontend"
    - In terminal, type "npm install"
    - In terminal, type "npm start"
7. Open project or navigate in web browser to: **http://localhost:3000/**

Required Test Scenarios

Create User/Login Test

- Scenario:
    1. Create three (or more) users.
    2. Two users will be linked to the same Restaurant.
    3. Log in and **click** "Generate Order" at least three times for each User.

- Purpose: Verify that the login system works and users can access the dashboard.

Order Display Test

- Scenario: With three users you created in the previous test.
    1. Log into each user mark two orders complete by "**double-clicking**" Order Card.
    2. View "Completed Orders" to view the orders you've completed.

- Purpose: Users linked to the same restaurant have access to view the same orders.
- Optional: **Click** the "Recall" button on an order in the Completed Orders list and this order will be "Pending" for all users linked to that Restaurant.

Change Restaurant Location

- Scenario:
    1. Click on (red) restaurant icon and choose "change address"
    2. Change address to whatever address you like and click confirm.

- Purpose: Customizable restaurant location.

Optional Test Scenario

View Analytics

- Scenario:
    1. Log into two or more accounts linked to separate restaurants.
        - Click "generate order" button at least ten times.
        - Double-click the orders one-by-one to complete them.
    2. View analytics for each User.

- Purpose: Verify that the analytics work for each User.