**Feb 25th, 2025**

**Retrospective**

What worked or went well?

* Design diagrams & sprint plan were clear and not overly complex
* Division of tasks over time ensured a balanced amount of effort across team
* Were able to resolve obstacles so that they didn’t hinder other members’ tasks for the week
* Completed user stories 1 & 2 as was the goal for the sprint

What caused problems, failed to work properly, or did not go well?

* Did not test for mobile view initially
* Had problems with original SQL stack, so ended up using a self-hosted server
* Had a few issues with Github as team members were learning how to use it
* Integration testing and implementation for Clerk was challenging to learn

What can be done differently in the next sprint to improve the process?

* Test all new UI elements on different view sizes
* Members experienced with elements of the stack will assist other team members who need to use the same tools

**Planning**

Any feedback from Brendan?

* Group is on a good pace and is following the scrum methods

Goals: Inventory tracking, sales tax, and expense report generation, create progress report 2

Delivery Plan:

* + Tasks for this sprint:
    - Researching how to integrate databases and unify CRUD operations between them
    - Researching the best sales tax practices and laws so that the software automatically makes users tax-compliant
    - Ensure each subsection is accessible and compatible with the dashboard
    - User Story 5 - Inventory
      * Create a database for the inventory
      * Attach/integrate with the income and expense databases
      * Use Jest and manual testing to ensure the integration works correctly with income/expense databases
    - User Story 4 - Sales Tax Collection + Remittance
      * Create a form to input sales tax that is sent to income and expense databases
      * Develop back-end functionality using endpoints to handle logic to fetch sales tax rate, income, and expense data from the database after which sales tax can be calculated
      * Utilize a library to format and generate reports from retrieved information
  + User Story 3 - Report Generation
    - Create a UI feature where users can select the type of report (income or expense) and specify the period for the same.
    - Develop back-end functionality using endpoints to handle requests for generating requested reports and logic to fetch the data from the database.
    - Utilize a library to format and generate reports and graphs from retrieved information.
    - Test the user story using Jest

Any comments or questions?