**Project Participants:**

Trevon Ross

**Title:**

Accounting WebApp

**Executive Summary:**

The Accounting WebApp is a project intended to provide accounting functionality through a web application. The goal is to allow users to manage products, create invoices, and track transactions. The web application will have a database design that includes entities such as Products, Invoices/Cart, and Invoice Items/Cart Items.

**Initial Features:**

* As a customer, I want to be able to add an item to an open invoice, so that I can purchase it.

POST /invoices/{invoice\_id}/items

* As a customer, I want to be able to modify the quantity of a product on an invoice so that I can get more or less as needed.

PUT /invoices/{invoice\_id}/{product\_id}

* As an owner, I want to be able to make an invoice as closed so that no more items can be added to a processed or completed order.

PUT /invoices/{invoice\_id}/close

/invoices/{invoice\_id} { state: 'fulfilled' }

*Example of list of Features/Endpoints for a Library API:*

* Product:
  + Create a new product (POST /products)
  + Retrieve all products (GET /products)
  + Retrieve a specific product (GET /products/{product\_id})
  + Update a product (PUT /products/{product\_id})
  + Delete a product (DELETE /products/{product\_id})
* Customer:
  + Create a new customer (POST /customers)
  + Retrieve all customers (GET /customers)
  + Retrieve a specific customer (GET /customers/{customer\_id})
  + Update a customer (PUT /customers/{customer\_id})
  + Delete a customer (DELETE /customers/{customer\_id})
* Invoice:
  + Create a new invoice (POST /invoices)
  + Retrieve all invoices (GET /invoices)
  + Retrieve a specific invoice (GET /invoices/{invoice\_id})
  + Update an invoice (PUT /invoices/{invoice\_id})
  + Delete an invoice (DELETE /invoices/{invoice\_id})
* CustomerInvoice:
  + Assign a customer to an invoice (POST /customer-invoice)
  + Retrieve all invoices for a specific customer (GET /customers/{customer\_id}/invoices)
  + Retrieve all customers for a specific invoice (GET /invoices/{invoice\_id}/customers)
  + Update the relationship between a customer and an invoice (PUT /customer-invoice/{customer\_id}/{invoice\_id})
  + Remove the relationship between a customer and an invoice (DELETE /customer-invoice/{customer\_id}/{invoice\_id})
* Invoice Item:
  + Add a product/item to an invoice (POST /invoices/{invoice\_id}/items)
  + Retrieve all items on an invoice (GET /invoices/{invoice\_id}/items)
  + Retrieve a specific item on an invoice (GET /invoices/{invoice\_id}/items/{invoice\_item\_id})
  + Update the quantity of an item on an invoice (PUT /invoices/{invoice\_id}/items/{product\_id})
  + Delete an item from an invoice (DELETE /invoices/{invoice\_id}/items/{product\_id})

**Stretch Goals (to be completed if time allows, or after graduation):**

* Generate financial reports: Implement the functionality to generate financial reports such as balance sheets, income statements, and cash flow statements.
* Payment integration: Integrate a payment gateway to enable online payment processing for invoices.
* Multi-user support: Implement role-based access control to allow different levels of access for different users (e.g., admin, accountant, customer).
* Integration with accounting software: Explore integration options with popular accounting software to synchronize data between the web application and external accounting systems.
* Automated invoice reminders: Set up automated reminders for overdue invoices and send notifications to customers.
* Data analytics: Implement data analytics features to provide insights into financial trends, sales patterns, and customer behavior.