**CAFFEINE LAMANNA MEAN BLOCK FULLSTACK**

Update code to position for Web 3.0 and bankless transition

With Git Repo and E2E testing framework

**FrontEnd:**

Store Front

Frontend client-side store front for coffee drinkers to order coffee to their PO from their web browser. Includes Braintree(traditional) and ZkSync(bankless) payflows with Angel Token Crowdfund capability.

About! – The Spiel

Buy Beans! - Inventory Request Service:

get green, packaged, discounted, merchandise inventory from backend database

Only shows non-zero inventory

Subscriptions:

Switch from Paypal Default to Braintree Custom w/ Paypal option

Angels Room (Bankless):

Connect to Metamask & display wallet address in corner

Display of tokenized funding rounds owned by customer

About pages for Last Round, This Round, and Next Round(s)

Click on token to go to token page in treasure chest

Crowdfund/ProfitShare Matic contract to buy tokens

TreasureChest:

View past crowdfund tokenized rounds by symbol grouping

View tokens individually

ShoppingCart:

Add items to local storage

hold items in session cart until checkout

change quantity and remove from cart buttons

Checkout Payflow:

Customer info: Billing Address/Shipping Address, Save in Vault for Easy Checkout

Promocode Request and Final Price adjustment

Payment Method:

Braintree:

Sandbox:

createNonce, hostedFields, sendPayment, return Info, Vault

Live:

credit cards, Paypal, Apple Pay, Google Pay, & Venmo

Zksync Bankless Checkout:

TestNet

MainNet

Checkout Complete:

Checkout page with confirmation info

BehindTheCounter

Frontend client-side Employee Interface:

About! – The Employee Spiel

Authorization Only:

Register Employee

Check Auth

Inventory: View All/Each, Post Green, Post Packaged, Post Merchandise

Promocodes: View All, Create new, Update,(payouts linked to smart contracts & wallets)

Roast List: A list of orders that need to roast and ship

Customer list in reverse

Only Shows What needs to be Roasted

Can Mark as Roasted – Moves to “Need to Ship” List

Purchase Orders List:

List of Orders newest to oldest,

Order functions(refund, change, etc.)

Customer List:

list of customers by name and email

Stats: shows totals, promocode usage, date started

Separate by Active and Inactive

Manifest Tokens:

Define parameters for a new funding round

Mint new tokens (bankless fund raising) via AngelTokens SmartContracts

New tokens get a new url and json file when manifested

TreasureChest:

View past crowdfund rounds by symbol grouping

View tokens individually

View more sensitive information/stats specific to token owner

(i.e. all time sales, total endeavors, total successful endeavors, etc….)

Links to other operation essentials:

Payment Processors, Post Office, Green Beans, ShipStation, Bag Supplier, SmartHome etc

Links to DeFi:

Zapper, Alchemix, Decentraland

**BackEnd:**

Routes to DB tables

Users:

Get a user, Post a New User, Update a User

Inventory:

Get Inventory, Post New Inventory, Update Inventory

CustomerDatabase:

Get Billing/Shipping, Post Billing/Shipping, Save information for easy checkout

Name, Email & Contact, Total Contributions, Order Details

Promocode:

Get a code, Post a New code, Update (tally promocode)

Checkout:

Sandbox & Live: Braintree, ZkSync

Checkout Complete:

Send Order to Database:

(insert\_into\_purchase\_orders.php) 🡪 translate to Node

Update inventory:

(update\_inventory\_after\_purchase.php) 🡪 translate to Node

Tally Promocode Usage:

(tally\_promo\_code.php) 🡪 translate to Node

Send to ShipStation

Send Customer Confirmation Email

PurchaseOrders:

Get order, Post new orders, Update orders

TreasureChest:

Get Token, Post New Token, Show Token JSON by ID Filename

MySQL

Auth, Inventory, Promocode, Order list, Customer list, Token Table (Treasure Chest)

Mongo

Local Smarthome Data

Solidity Contracts

Alms

AngelTokens

AngelTokens Execution

Truffle to migrate & deploy crowdfund contracts

**Tests and code watchers**

**E2E tests in Jest**

**\*\* StoreFront \*\***

Routes:

Test all Routes for 200 statuses

Test for proper user exclusion

Checkout Flow:

Test checkout flow forms: addressForm, promoForm, payForm, confReturn

Test Sandbox Gateways & testNets for affirmative and error cases

Database Checks:

Monitor Databases for proper function and variable type

Test Database Entry affirmative and error cases

**\*\* BehindTheCounter \*\***

Routes:

Test all Routes for 200 statuses

Test for proper user exclusion

Database Checks:

Monitor Databases for proper function and variable type

Test Database Entry affirmative and error cases

**\*\* SmartHome \*\***

Routes:

Test all Routes for 200 statuses

Test for proper user exclusion

Database Checks:

Monitor Databases for proper function and variable type

Test Database Entry affirmative and error cases

**Full Cup (the hail mary)**

Connect smarthome to CafLaM Fullstack

Integrates:

Energy Production/Consumption

Home business POS system

Home, Garden, and Business Inventory & Database

With:

StoreFront

Backend Employee UI

Client Side PayFlow

Bankless Fund Raising & Payments

Blockchained Inventory & Assets

Defi Deleveraging and portfolio management

Discord channel to talk about it all

\*\*Known Issues:

Crowdfund execution contract

Checkflow CSS is not right on OrderBeans page

should be center screen in front and on top

background should fade

should render like on store\_front.html