

# Future-Proofing Physical Retail with AI and IoT



SAM BROCKLEHURST

**VEND-E** Tuesday, 27 April 16:37

**Live Occupancy** 26 / 40

**Trending - Last hour** 10 interactions  
Quaker Oats So Simple

**Underperforming - Last hour** 1 interactions  
Adidas Supercourt White

**Price Match**  
Our Price £20.00 Amazon £17.99 -11%  
Sneak Energy Bottle Green

**Products**

Name	Interactions	Price	Price Matched?	Competitor Prices
Nestle Nesquik Cereal	14,313	£2.00	✓ 0%	ASDA £2.00, Sainsbury's N/A
Kellogg's Coco Pops	13,100	£1.55	✓ +25%	ASDA £1.50
Bulk Powders Protein	13,708	£13.18	✗ -10%	£11.99
Sneak Energy Bottle Green	13,689	£20.00	✗ -11%	£17.99
Quaker Oats So Simple	14,089	£1.80	✓ +72%	£3.40
Nestle Shreddies	14,013	£2.40	✓ 0%	£2.40
Adidas Supercourt White	13,852	£54.00	✓ +2%	£55.00
Nestle Golden Nuggets	13,743	£2.10	N/A	N/A

**Trending - Today**

Name	Interactions
Quaker Oats So Simple	85
Nestle Nesquik Cereal	82
Nestle Golden Nuggets	61
Adidas Supercourt White	57
Sneak Energy Bottle Green	53

**Underperforming - Today**

Name	Interactions
Adidas Supercourt White	57
Nestle Golden Nuggets	45

# INTRODUCTION



Research

Methodology

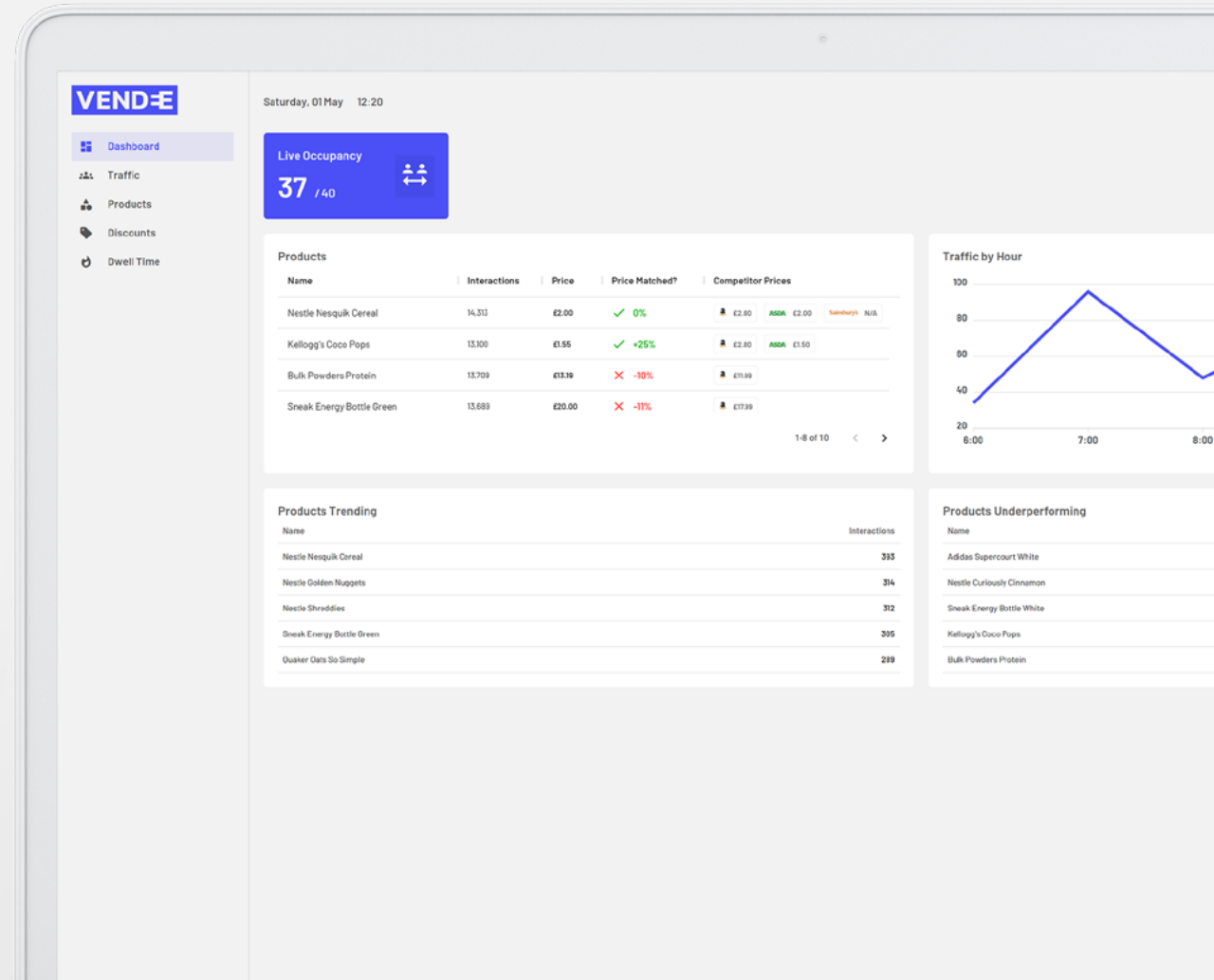
Design

Implementation + Testing

**VEND-E** Demo

Evaluation

Conclusion



# Current Retail Landscape

**32.4%**

US E-Commerce Grown in 2020

E-COMMERCE



**63%**

UK Retail E-Commerce Sales in 2020 from Smartphones

M-COMMERCE



**4/10**

Consumers Switched to Online for Products They Would Have Bought Offline

COVID-19



# The Role of Physical Retail



## SHOWROOMING

When consumers inspect products offline, before making the purchase online.

## EXPERIENCE

This product research is still valuable to retailers, providing customers with the chance to discover their products.

## HOW TO MEASURE STORE PERFORMANCE?

Sales and profits are no longer suitable, as value is still being created through the pre-transaction stage of the customer experience. New metrics are needed:

| Customer Behaviour

| Product Engagement

| Dwell Time

# Omnichannel Retailing

A strategy that focusses on providing a seamless shopping experience, satisfying customers who increasingly want everything.

However, there are problems for retailers:

| **Data Integration**

| **Lack of Technology**



# Technologies

Capturing and utilising data in the physical space is essential for omnichannel retailers.

| Machine Learning Model

| Facial Recognition

| Object Detection

ARTIFICIAL INTELLIGENCE



| Cameras

| Digital Screens

| BLE Beacons

INTERNET OF THINGS



# Product



## AIM

To develop a system that enables retailers to fully integrate omnichannel strategies by capturing and utilising data in the physical retail space.

- Automatically capture customer behaviour and engagement in real-time.
- Utilise data across the store.
- A dashboard to provide retailers (managers) with an overview of store performance.

**MUST** ✓

- Branding to make it more marketable to potential retailers.
- Enable personalised content to be sent to customers.
- Offer dynamic pricing on products.

**SHOULD** ✓

- Perform automatic checkout when customers leave the store.
- Track when products in-store are out of stock.

**COULD** ✓

- A companion smartphone app to enhance customer experience in-store.

**WON'T** ✗

# Agile

A modern, iterative approach to project management and software development.

PRODUCT BACKLOG



USER STORIES



USER PERSONAS



UNIT TESTING





# Product Specification

Produced to define the product requirements and design, including hardware and functions.

### User Personas

Persona 1

**Bio**  
Emma operates a small grocer in London. Her shop has seen declining sales over the past year, with the COVID-19 pandemic, this sales have all but disappeared. She fears that her loyal customers have been lost to the giant, online retailers who could continue to operate during the pandemic.

**Needs**

- Win back her customers in the post-pandemic economy.
- Cut costs in-store through requiring less staff.
- Offer competitive prices, so her customers don't feel forced to shop online.
- Reward her loyal customers.

**Tech**

- IT
- Software
- Mobile Apps
- Online Shopping

**Frustrations**

- Following strict government regulations on social contact.
- People using her store's resources without reciprocating with a purchase.

### User Stories

- As a store manager, I want to see what times of day my shop is busiest so that I have the correct amount of staff available.
- As a customer, I want to be automatically charged for products I pick up so that my shopping trip is quicker and more convenient.
- As a replenishment assistant, I want to be able to see when stock is low so that I can quickly refill the shelves.
- As a store manager, I want to know what my competitors prices are so that I can ensure I am offering a competitive price.
- As a store manager, I want to see what products are underperforming so that I can adjust their prices and position them around the store to increase sales.
- As a customer, I want to receive loyalty discounts so that I can be rewarded for shopping in store.

### Wireframes

1 Dashboard

2 Traffic

### Hardware

Several hardware devices will be used to capture in-store data and utilise it:

- Video cameras used to capture customer behaviour and engagement in real-time
- Beacons to broadcast personalised content to nearby customers
- Digital screens to dynamically update product prices

These devices need to communicate and synchronise with the back-end. This is enabled with a Raspberry Pi, a single-board computer with 4G general-purpose input/output pins (GPIO), suitable for interfacing with the devices.

A store management app is essential for store managers to generate valuable insights from the data captured, and allowing them to see store traffic, manage products, and personalise the customer experience. The app will be developed using React, a JavaScript library for building user interfaces, and Electron, a framework for creating desktop apps. MongoDB will be used for the database, as it works well with:

### Technologies

The product will benefit from ML, to automate the process of identifying and observing customers in-store and the products they engage with. TensorFlow (TF) is an open-source ML platform allowing developers to build and train models, and to use them in various production environments. For this product, TF can enable video cameras to detect customers and recognise the individual products they interact with. For this to work, there needs to be a back-end to interact with the devices, to access the video feed and run it through the TF model, and to save and output the data. Node.js is an appropriate technology for interacting with the Raspberry Pi and the devices connected to it through the GPIO pins. As it is mostly used for server-side scripting, Node.js can also serve this data to the front-end.

A store management app is essential for store managers to generate valuable insights from the data captured, and allowing them to see store traffic, manage products, and personalise the customer experience. The app will be developed using React, a JavaScript library for building user interfaces, and Electron, a framework for creating desktop apps. MongoDB will be used for the database, as it works well with:

### Functional Specification

Vend's functionality can be broken down into the following sections, starting with each page of the app:

**All Pages**

- A navigation component that switches between pages.
- A header that shows the current date and time, updating each second.

**Dashboard**

- Displays the main component from the other pages to provide an overview of the most crucial data.
- Top panels should include the live occupancy component.

**Traffic**

- Display graphs for store traffic per hour and per week.
- Graphs should be interactive - hovering over them will reveal more data.
- Graphs should be reusable for the current day, but allow users to select the date/time to see past data.
- Top panels should show the live occupancy component and the day's traffic - this should show a percent change from the previous day.

**Customers**

- List all customers, with sort and filter functions.
- List all customers in-store.
- Customer list should have a column for customer loyalty.
- A button that opens a dialog box for adding a new discount.
- Discounts can be attached to all products, or an individual product.
- Discounts can be attached to all customers, or an individual set of customers.
- Top panels should show the current discounts - clicking it will reveal the customers affected by it in a dialog box, with the ability to edit any values.

**Products**

- List all products, with sort and filter functions.
- Display stock level of each product as a column.
- Dynamic pricing as columns, with each of the competitor prices being listed and the percentage difference between the current price and the lowest competitor price.

# Branding

Modern and sleek brand assets to help the product distinguish the product from its competitors.

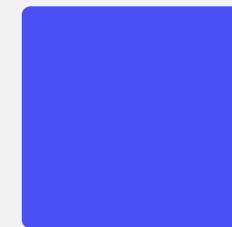
Logo



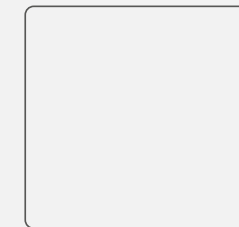
Icon



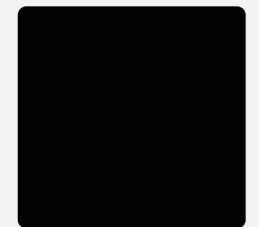
Colours



#4a4ff6



#f2f2f2



#010300

# Initial Setup



## HARDWARE

Test Pi + Camera

Access Pi from Local Machine

Get Camera Feed



## MACHINE LEARNING

Integrate Pre-Trained Model

Start Detecting Objects



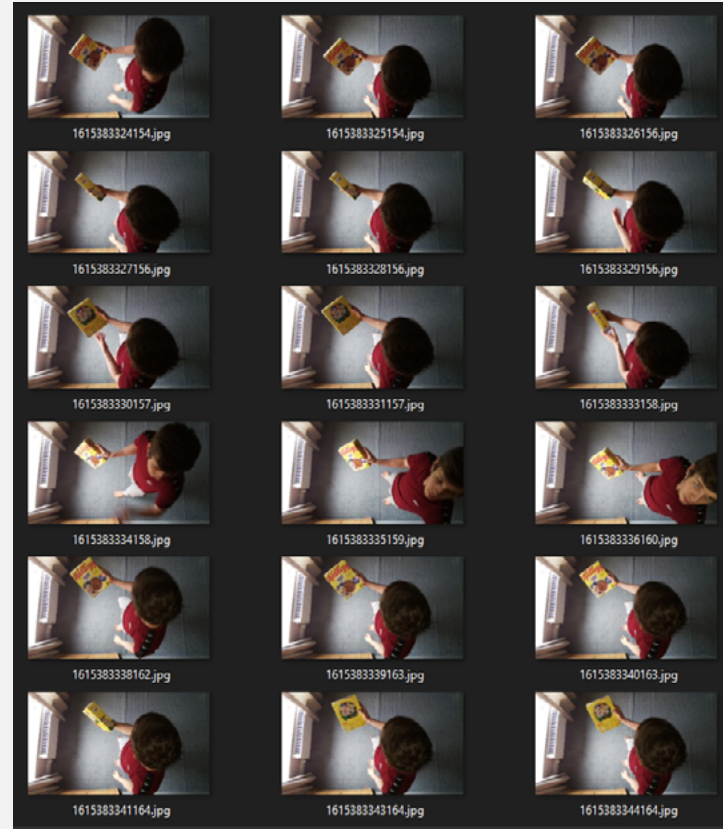
```
Cmder
{
  {
    score: 0.8401,
    classId: 1,
    class: 'person',
    bbox: [ 0, 0.2129, 0.9232, 0.8057 ]
  },
  {
    score: 0.6777,
    classId: 77,
    class: 'cell phone',
    bbox: [ 0.2042, 0.2448, 0.3733, 0.3554 ]
  },
  {
    score: 0.5669,
    classId: 84,
    class: 'book',
    bbox: [ 0.6047, 0.1126, 0.7681, 0.2916 ]
  }
}
}
C:\Projects\vendee\detect (main -> origin)
λ
```

# Custom Machine Learning Model

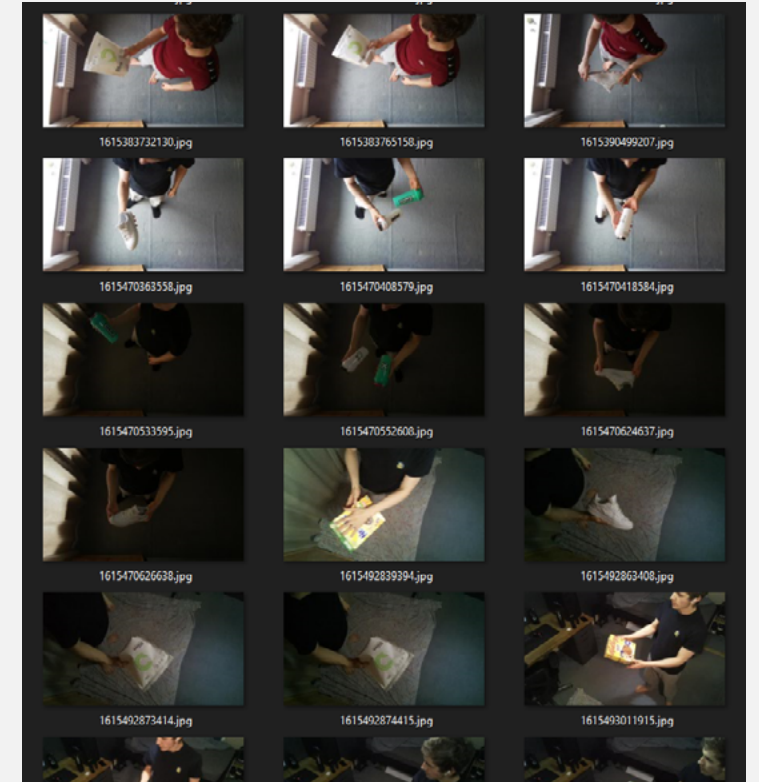


## PREPARING DATASET

Various datasets were produced to train a robust model, capable of detecting people and retail items.



1<sup>st</sup> Dataset



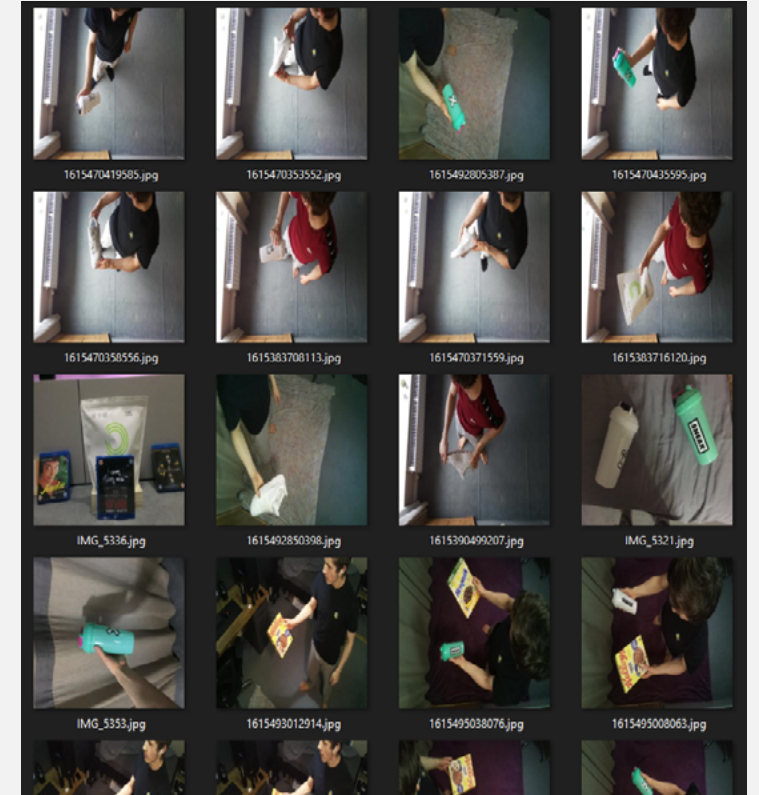
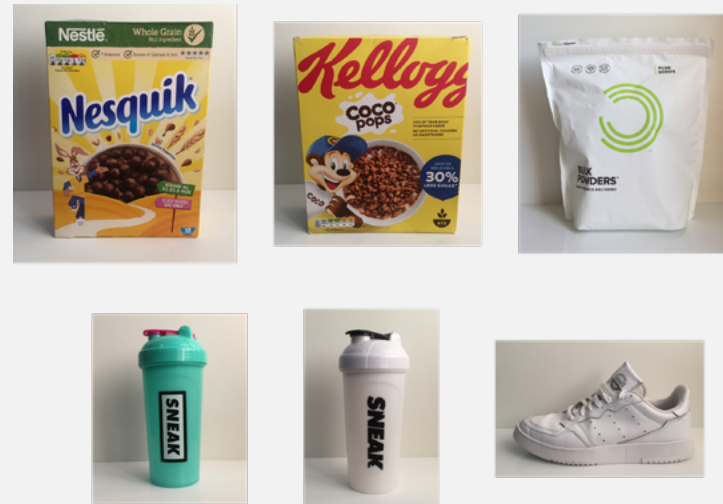
2<sup>nd</sup> Dataset

# Custom Machine Learning Model



## PREPARING DATASET

Various datasets were produced to train a robust model, capable of detecting people and retail items.



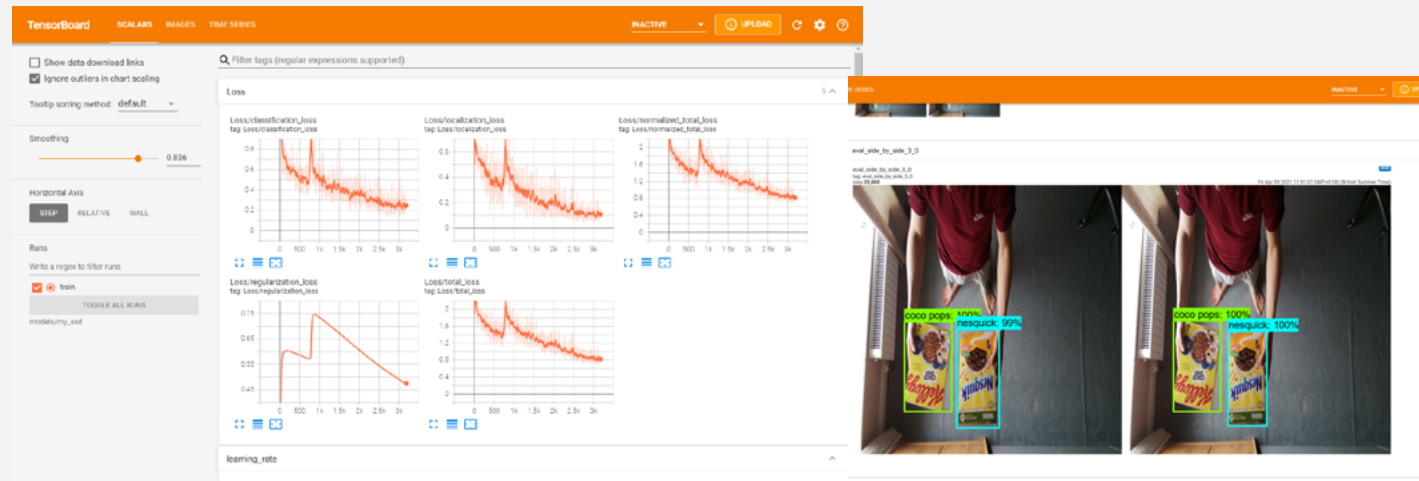
Final Dataset

# Custom Machine Learning Model



## TRAINING

The dataset is used to train the model to accurately predict the retail products.



```

Administrator: Anaconda Prompt (Anaconda3) - python model_main_tf.py --model_dir=model/my_csd_resnet152_v1_tf --pipeline_config_path=
I0307 22:44:28.646136 8064 model_lib_v2.py:665] Step 6200 per-step time 0.438s loss=38.321
INFO:tensorflow:Step 6300 per-step time 0.426s loss=37.245
I0307 22:45:11.441008 8064 model_lib_v2.py:665] Step 6300 per-step time 0.426s loss=37.245
INFO:tensorflow:Step 6400 per-step time 0.401s loss=36.132
I0307 22:45:52.896247 8064 model_lib_v2.py:665] Step 6400 per-step time 0.401s loss=36.132
INFO:tensorflow:Step 6500 per-step time 0.408s loss=35.153
I0307 22:46:34.501177 8064 model_lib_v2.py:665] Step 6500 per-step time 0.408s loss=35.153
INFO:tensorflow:Step 6600 per-step time 0.395s loss=34.028
I0307 22:47:16.626177 8064 model_lib_v2.py:665] Step 6600 per-step time 0.395s loss=34.028
INFO:tensorflow:Step 6700 per-step time 0.400s loss=33.082
I0307 22:47:58.141617 8064 model_lib_v2.py:665] Step 6700 per-step time 0.400s loss=33.082
INFO:tensorflow:Step 6800 per-step time 0.419s loss=32.529
I0307 22:48:39.651280 8064 model_lib_v2.py:665] Step 6800 per-step time 0.419s loss=32.529
INFO:tensorflow:Step 6900 per-step time 2.228s loss=31.343
I0307 22:50:49.094719 8064 model_lib_v2.py:665] Step 6900 per-step time 2.228s loss=31.343
INFO:tensorflow:Step 7000 per-step time 0.411s loss=30.469
I0307 22:51:47.962148 8064 model_lib_v2.py:665] Step 7000 per-step time 0.411s loss=30.469
INFO:tensorflow:Step 7100 per-step time 0.428s loss=29.826
I0307 22:52:37.285154 8064 model_lib_v2.py:665] Step 7100 per-step time 0.428s loss=29.826
INFO:tensorflow:Step 7200 per-step time 0.443s loss=28.696
I0307 22:53:18.923472 8064 model_lib_v2.py:665] Step 7200 per-step time 0.443s loss=28.696
INFO:tensorflow:Step 7300 per-step time 0.421s loss=27.911
I0307 22:54:00.666702 8064 model_lib_v2.py:665] Step 7300 per-step time 0.421s loss=27.911
INFO:tensorflow:Step 7400 per-step time 0.428s loss=27.151
I0307 22:54:43.370002 8064 model_lib_v2.py:665] Step 7400 per-step time 0.428s loss=27.151
INFO:tensorflow:Step 7500 per-step time 0.413s loss=26.527
I0307 22:55:26.040978 8064 model_lib_v2.py:665] Step 7500 per-step time 0.413s loss=26.527
INFO:tensorflow:Step 7600 per-step time 0.403s loss=25.714
I0307 22:56:08.812669 8064 model_lib_v2.py:665] Step 7600 per-step time 0.403s loss=25.714
    
```



# Technologies In-Store

## \$ \_\_\_\_\_

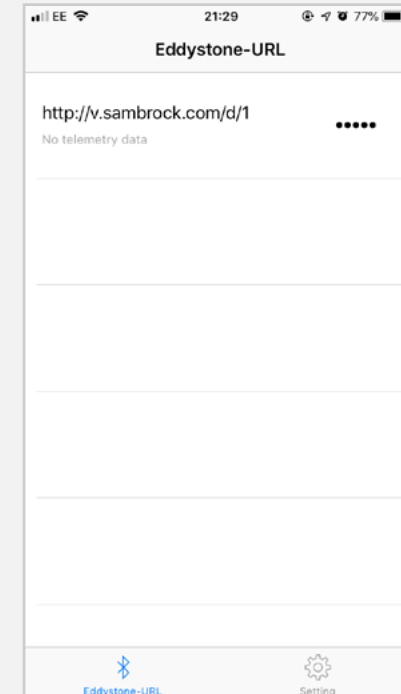
### DYNAMIC PRICING

- | LCD Screens Displaying Price
- | Competitive Price Matching

## 🏷️ \_\_\_\_\_

### DISCOUNTS

- | Bluetooth Beacons
- | Eddystone-URL



## IMPLEMENTATION + TESTING

# Web App



## FRONT-END

React

Jest

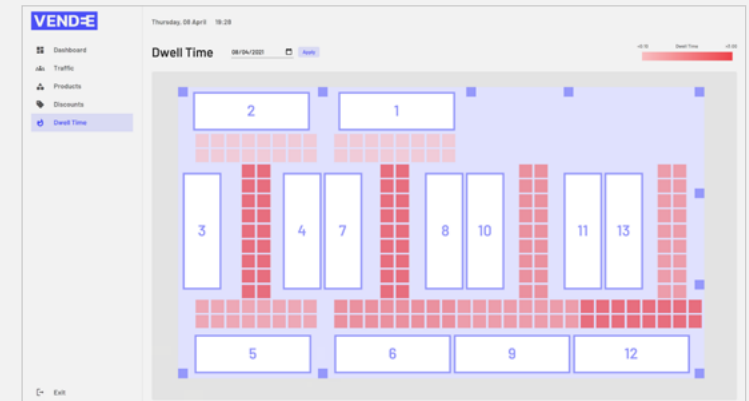
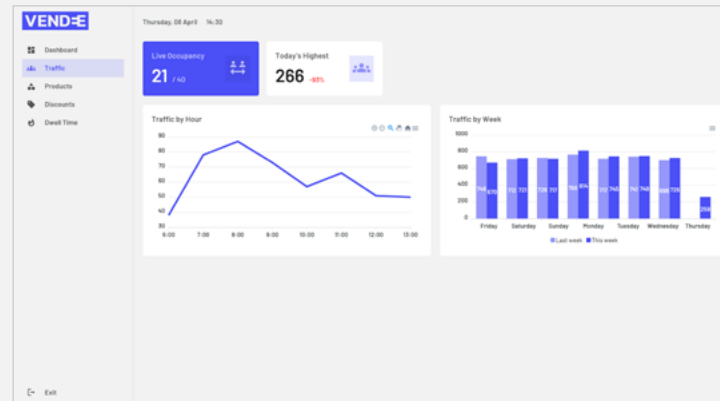
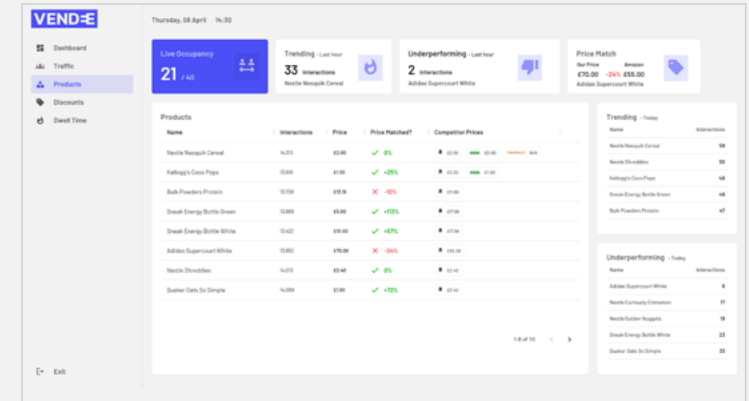
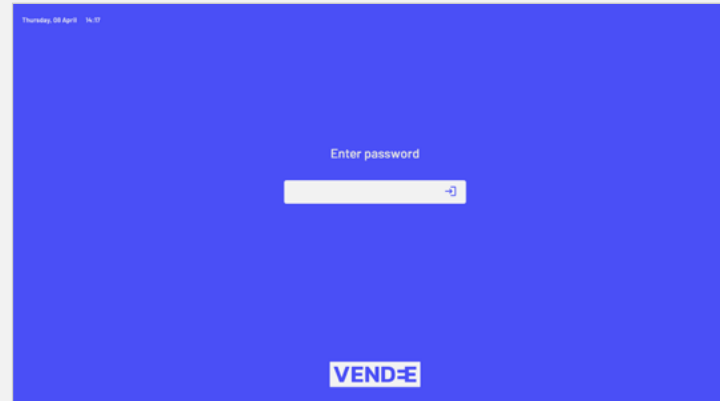


## BACK-END

Node.js

Express

MongoDB







**VENDEE** Demo

# Future Developments

Vendee has many opportunities for future development.

Companion App



Customer Facial Recognition



Progressive Web App



More Products in Dataset



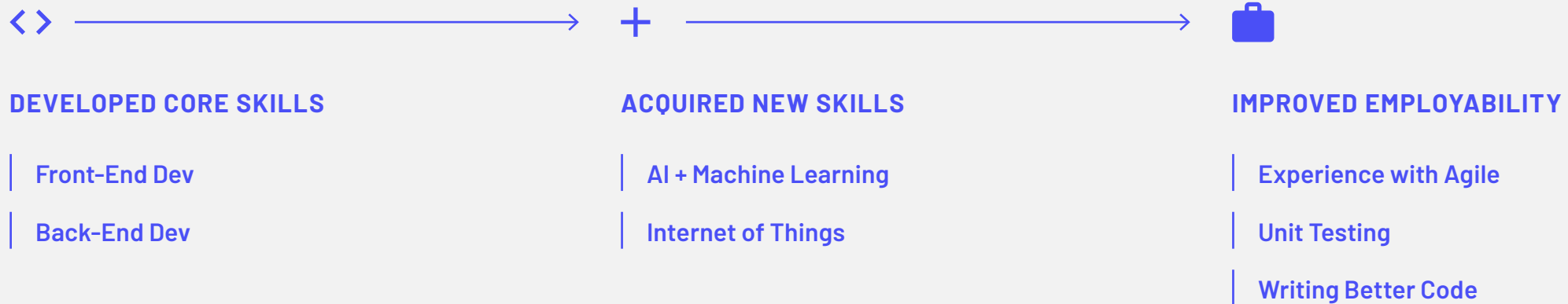
Custom Hardware



Access Levels



# Learning and Self Development



## CONCLUSION



# Future-Proofing Physical Retail with AI and IoT

Post-Pandemic Retail Landscape



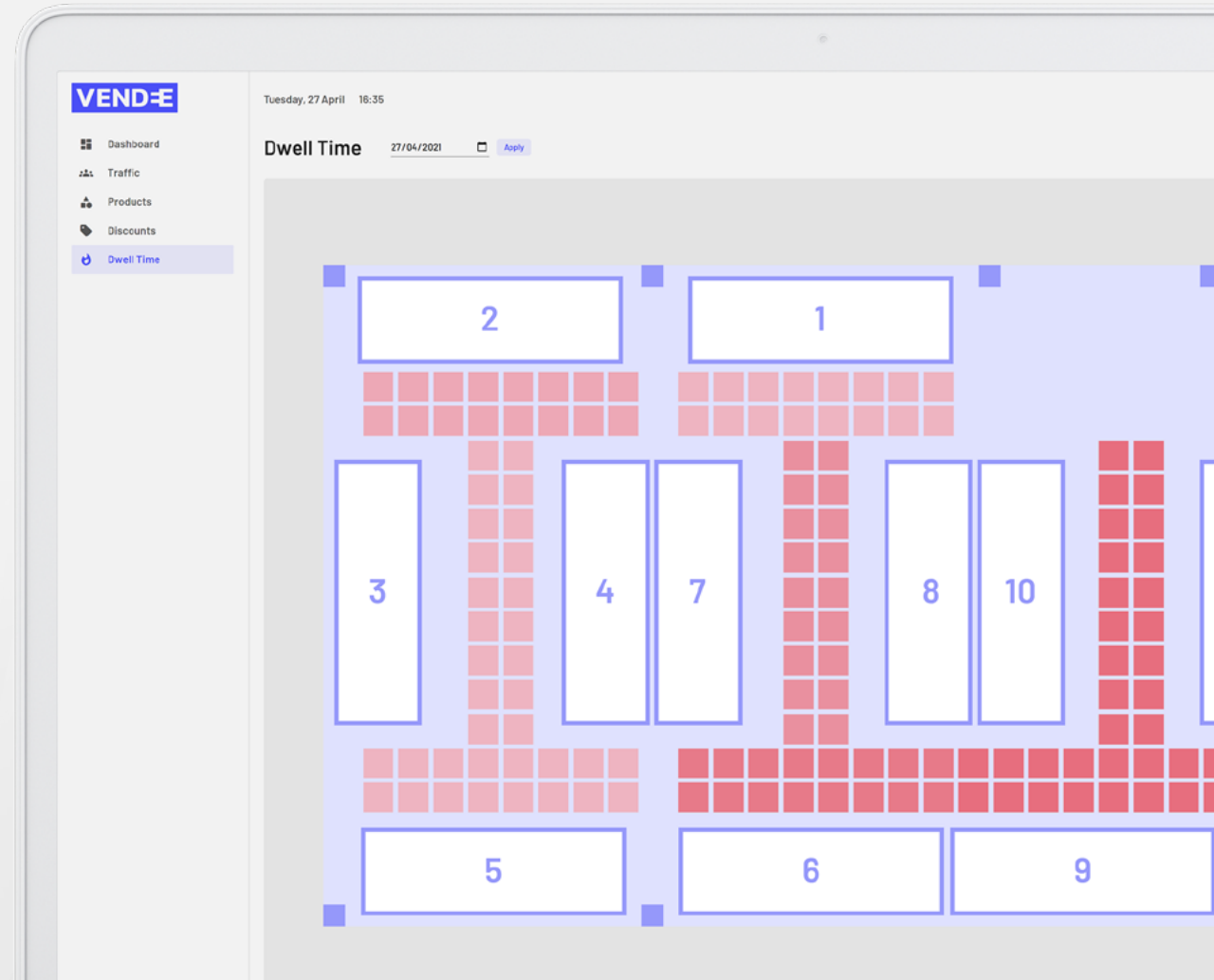
Omnichannel Strategy



AI + IoT



Viable Solution



A blue-tinted photograph of a grocery store aisle. The aisle is filled with various products, including bags of produce and boxes of goods. Price tags are visible, with one prominently displaying "SAVE" and "R23.99". Another tag shows "SAVE" and "R24.99". The background shows more shelves and a "DOOR SALE" sign. The overall scene is a typical retail environment.

**Thank you for listening.**