NUS
National University
of Singapore

CP2106 (Orbital)

Peh Hoe Khim Marcus
Ee Kar Hee, Nicholas

Team Name:
Markeet

Level of Achievement:
Artemis

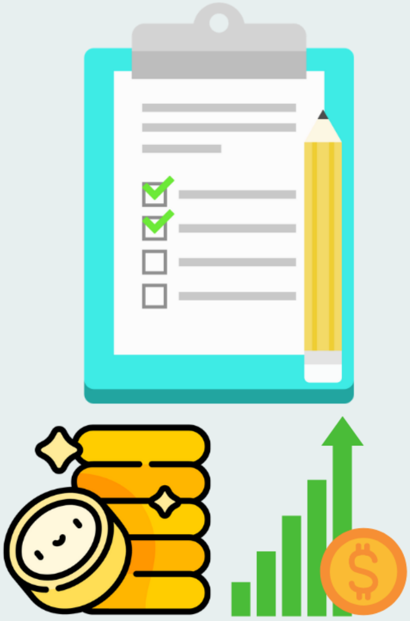
Year:
2021

TABLE OF CONTENTS

Poster

SALES & INVENTORY MANAGEMENT AT YOUR FINGERTIPS!

markeet



Developed with

- ✔ Sales Report
- ✔ Inventory Tracking
- ✔ Cashier Interface
- ✔ Telegram Bot Integration
- ✔ No ads

ABOUT US

Free-to-use, web-based Point of Sale system with real-time detailed sales report and inventory count.

Telegram integration allows customers to easily check store locations and stock inventory.

Scan here to find out more

Orbital'21 Markeet

Marcus & Nicholas

Code Base

https://github.com/marcuspeh/Markeet

Deployment

https://markeet.herokuapp.com/

Motivation

Ever run a small shop on Carousell or a flea market store but have no idea how to keep track of inventory and profit? Or perhaps you want to find out easily how much profit you are making? Enter markeet, a free Point Of Sale (POS) system that can help you solve the issue.

Or are you a customer, traveling down to a shop to buy a certain item, only to find out the item is out of stock. If the shop uses markeet, time spent travelling to the shop can be saved too. You can easily check the stock of the item on your phone and even find the nearest shop that sells the item.

Aim

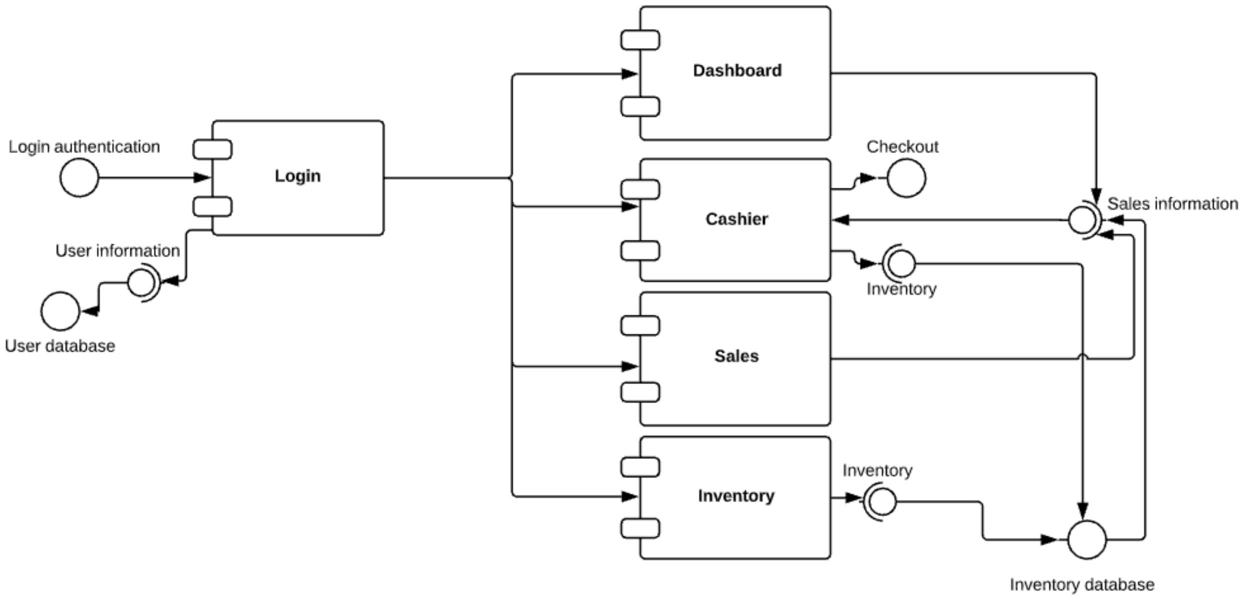
The aim of the project is to benefit small retail shops by creating an open source and free to use Point Of Sales (POS) system. Many small businesses currently use traditional methods to keep track of sales and conduct stock-taking, be it due to budget constraints or the small scale of the business. As such, there is value in implementing a POS system that can be used easily anywhere at almost no cost. All that’s needed is an internet connection and a device to access the internet.

Customers can also check inventory count for the item, instead of wasting a trip down to the shop only to realise that it is out of stock. They can also locate the nearest shop that has stocks for the item they are looking to buy.

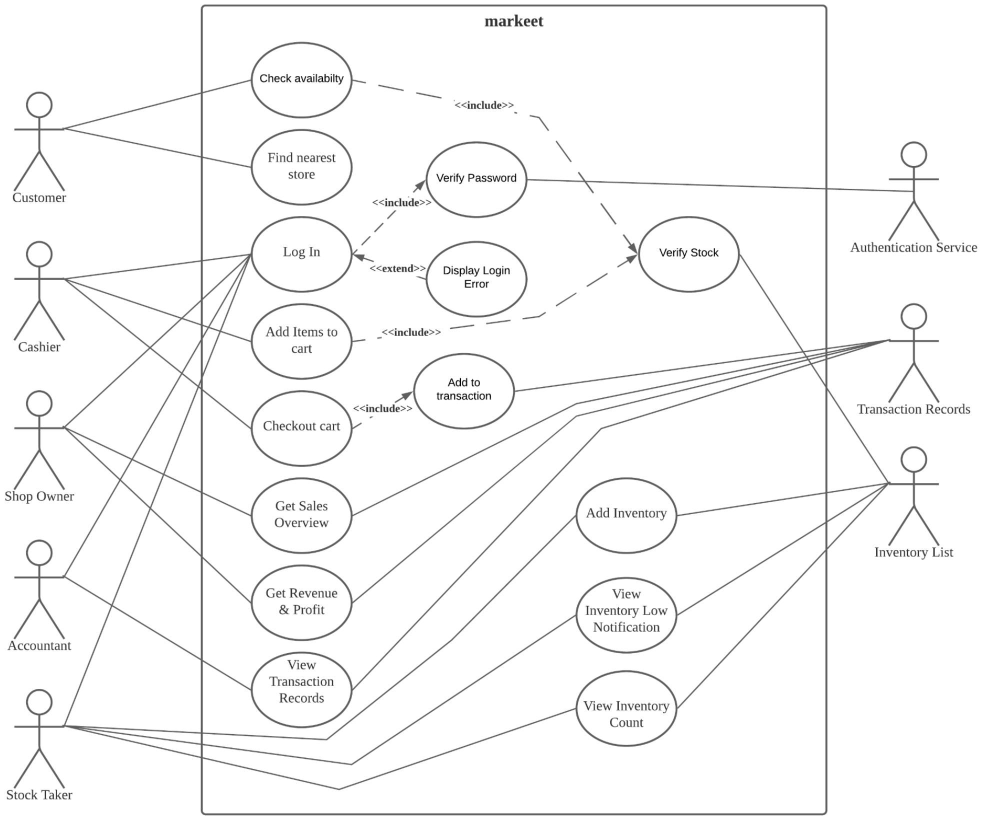
User Stories

1. As a stock taker for a shop, I can view the suppose stock in the shop so that i can double check the amount in the shop with ease
2. As a stock taker for a shop, I can view the notifications for low inventory so that I can restock in time.
3. As a stock taker for a shop, I can edit the inventory count for certain items so that the information in the app is updated.
4. As a stock taker for a shop, I can add in item(s) to inventory so that the inventory in the app is updated.
5. As a small business owner, I can get data from the monthly sales such as best selling items so that I can help my business grow.
6. As a business owner with multiple outlets, I can find out which shops are making a profit or loss so that I redistribute manpower.
7. As a cashier, I can use that cashier app that is easy to use and convenient so that I can clear the queue faster.
8. As a cashier, I can check the inventory so that I can answer customers' enquiries on stock.
9. As an accountant, I can check the sales of the shop easily on a monitor so that I can save time from not looking through paper and pen
10. As a customer, I can use the telegram bot to check for stock in a shop so that I can save time from travelling to the shop if it is not available.
11. As a customer, I can find the nearest shop that has the item I want to buy in stock so that I can save time travelling.

Component Diagram



Use Case Diagram



Features and Timeline

A **Web-based Point Of Sale (POS) system** that is free to use. It will help businesses generate real-time detailed sales reports and inventory count, on top of normal cashier operations.

The **Telegram Bot** provides a quick way for customers to check for stocks in a particular store and locate the nearest shop with the item they want to buy. The Telegram Bot will share the same database as the POS system

Core features:

(1) **Dashboard page** - The main page when a user logs in to the website. They will be able to see details on their shop, like inventory count, latest sales and other notifications. Links to other pages will also be available on that page.
(2) **Cashier page**- Where cashiers can enter sales in
(3) **Inventory page** - For live tracking of inventory. Options to add in inventory count and remove inventory will be available too. Updating inventory by CSV will also be supported.
(4) **Sales report page** - To see the profit/loss of the shop as well as the best/worst selling item. Other details regarding such as a breakdown of sales per day etc will also be shown.
(5) **Telegram Bot** - For customers to find out if the item is in stock. They can also locate the nearest shop with the items they are looking to buy.

Features to be completed by Milestone 1 (31 May)

- Design of the system
- Integrated frontend and backend system with a login feature
- (1) “Settings” page for changing settings
- (2) Barebones cashier page
- (3) Inventory list displaying all the inventory in the database
- (3) Ability to add in new stocks and sending it to the database

Features to be completed by Milestone 2 (28 Jun)

- (1) Display information about the shop
- (2) Entering of sales
- (3) Sorting of inventory list based on different method
- (3) Deduction of stock based on sales
- (3) Alert when stocks are running low
- (4) Ability to see basic sales report such as profit/lost
- (4) Sales information will be updated once sales are entered
- (5) Telegram bot basic layout and menu

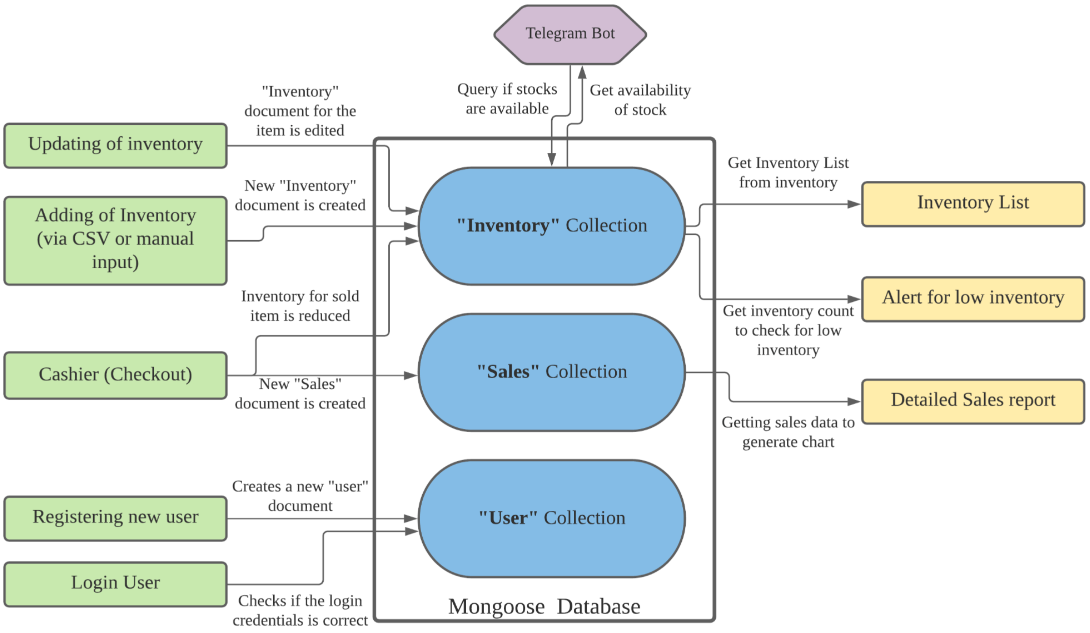
Features to be completed by Milestone 3 (26 Jul)

- (1) Add in options for settings such as Telegram bot support
- (3) Prediction of how much stock to order to replenish
- (4) Charts based on sales
- (5) Telegram bot support for customers to check stock
- (5) Telegram bot support for finding the nearest shop with stock

Possible future feature implementations

- Machine learning Telegram Bot
- Introductory tutorial page for newcomers
- Support for multiple stores per account
- Employees account for the shop

Outline


Telegram Bot
Query if stocks are available
Get availability of stock
Updating of inventory
"Inventory" document for the item is edited
Adding of Inventory (via CSV or manual input)
New "Inventory" document is created
Inventory for sold item is reduced
Cashier (Checkout)
New "Sales" document is created
Registering new user
Creates a new "user" document
Login User
Checks if the login credentials is correct
"Inventory" Collection
"Sales" Collection
"User" Collection
Mongoose Database
Get Inventory List from inventory
Inventory List
Alert for low inventory
Get inventory count to check for low inventory
Detailed Sales report
Getting sales data to generate chart

Timeline

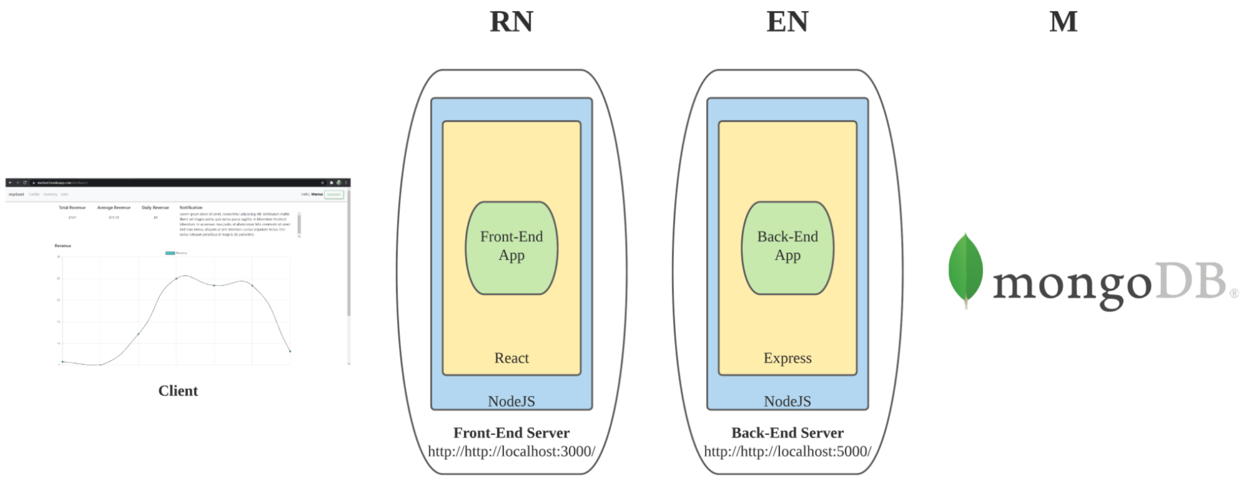
Dates	To-dos	Done
10 May - 16 May (Lift Off)	Complete ideation and architecture design	✓
17 May - 23 May	Build minimal system with login	✓
24 May - 30 May	Implement inventory	✓
31 May (Milestone 1) - 13 Jun	Implementing of core features	✓
14 Jun - 20 Jun	Complete the basic CRUD features	✓
21 Jun - 27 Jun	Debugging and basic tests	✓
28 Jun (Milestone 2) - 11 Jul	Integration of database with Telegram Bot	
12 Jul - 18 Jul	Further testing and debugging	
19 Jul - 25 Jul	Polishing up UI and features	
26 Jul (Milestone 3) - 25 Aug	Refinement	

Wire Frame

https://www.figma.com/file/mmQDZh2OWd4JVPEAiMECol/markeet?node-id=0%3A1

Tech Stack

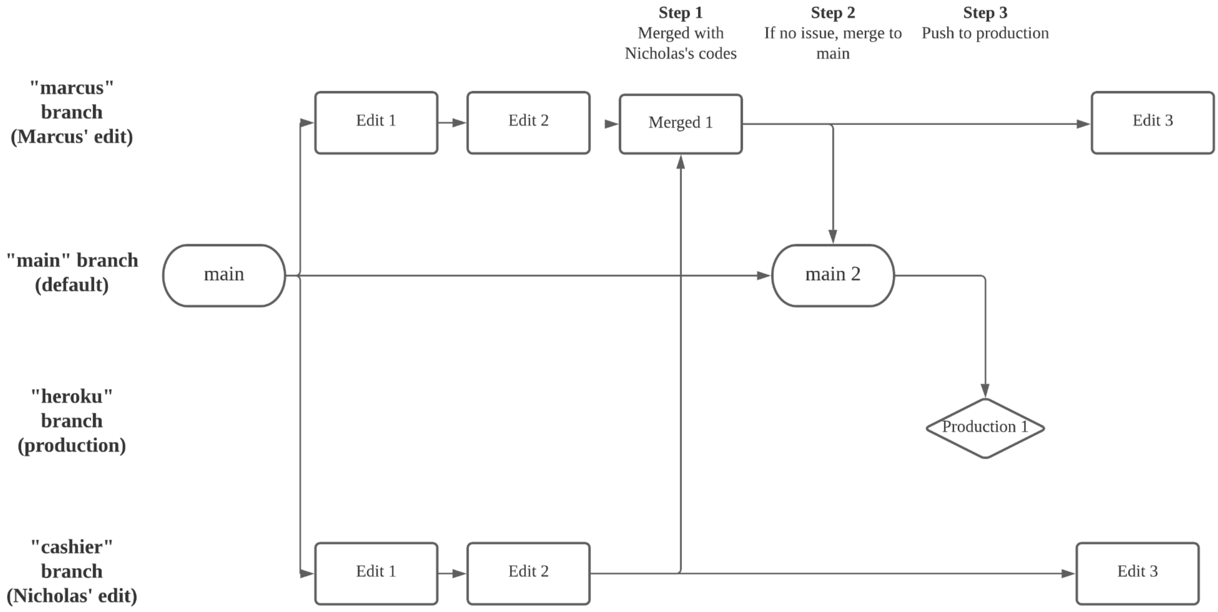
1. MERN Full stack (for website)
    a. MongoDB
    b. Express
    c. React
    d. Node JS
2. Redux (for website)
3. Telegram API (for Telegram Bot)
4. Python (for Telegram Bot)
5. Git & GitHub (for version control)
6. Jest (for integration testing)



Github workflow

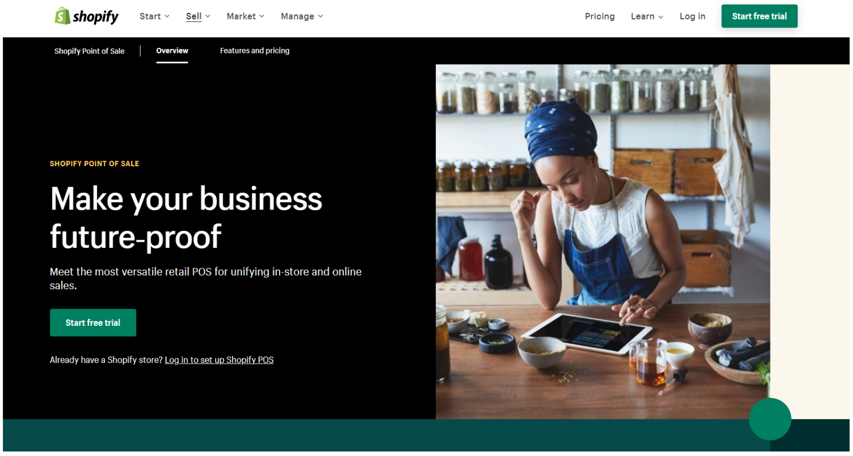
For our workflow in github, we are creating 4 branches.”main” for default, “heroku” for production, “cashier” for Nicholas and “marcus” for Marcus. We will be doing our codes in our individual branch. Before we push to main, we will be merging the other person’s codes into our branch to test. If there is no issue, our codes will then be merged into “main” and subsequently, “main” to be merged into “heroku” for production.

Here’s a flowchart visually our workflow.


Step 1
Merged with Nicholas's codes
Step 2
If no issue, merge to main
Step 3
Push to production
"marcus" branch (Marcus' edit)
"main" branch (default)
"heroku" branch (production)
"cashier" branch (Nicholas' edit)
main
Edit 1
Edit 2
Merged 1
Edit 3
main 2
Production 1
Edit 1
Edit 2
Edit 3

Comparison with other apps

Shopify POS (https://www.shopify.com.sg/pos)



SHOPIFY POINT OF SALE

Make your business future-proof

Meet the most versatile retail POS for unifying in-store and online sales.

Already have a Shopify store? Log in to set up Shopify POS

While shopify might be a very big ecommerce website and its POS system is no doubt one of the most user friendly additional services, its POS system is only limited to shopify users. In addition, shopify POS does not support Android devices but only IOS devices. It is also quite expensive to use its full features. Markeet, on the other hand, is completely free to use while enjoying most of the features present in shopify's POS. The most important aspect of markeet is the ability to be used on any devices, regardless of Android, IOS or windows.

Testing

We are using Jest (https://jestjs.io/) in order to write unit testing for react.

The testing mainly focus on the following:

1. **User input**. We tested if each button is working as expected. For example, clicking on the "inventory" button actually changes the screen to the inventory page instead of others.
2. **Entry input**. We tested possible user input to test if the webapp is performing as expected. For example, when registering as a new user, if the user enters an invalid email like "qwerty" instead of "qwerty@domain.com", the website should actually prompt as an error

Summary of test

Feature	Things to test	Test	Expected	Result
Landing	Able to see the landing page	Trying to go to the landing page when logged in	User will be redirected to dashboard page	✔
		Trying to go to the landing page when not logged in	User will be able to see the landing page	✔
	Able to register for an account	Opening the "register new user" page	Page should load properly	✔
		Registering with all empty fields	All the fields should show various error messages. *Name:* "Name field is required" *Email:* "Email field is required" *Password:* "Password field is required"	✔
		Registering with invalid email eg "qwerty"	Error message will be shown "Email is invalid"	✔
		Registering with password that is too short	Error message will be shown "Password must be at least 6 characters"	✔
		Password entered does not match	Error message will be shown "Passwords	✔

			must match”	
		Registering with an email that is already registered	Error message will be shown “Email already exists”	✓
		Registering with all valid inputs	Account will be registered	✓
	Able to log in	Logging in with none of the fields filled in	All the fields should show various error message *Email:* “Email field is required” *Password:* “Password field is required”	✓
		Logging in with an invalid email eg “qwerty”	Error message will be shown “Email is invalid”	✓
		Password does not match email	Error message will be shown “Password incorrect”	✓
		Email is not registered	Error message will be shown “Email not found”	✓
		Logging in with correct login credentials	Dashboard will be loaded	✓
Navbar	Navigation bar works as planned	Dashboard button brings the user to dashboard page	Dashboard page loads when clicked	✓
		Cashier button brings the user to cahier page	Cashier page loads when clicked	✓
		Inventory button brings the user to inventory page	Inventory page loads when clicked	✓
		Sales report button brings the user to sales report page	Sales page loads when clicked	✓
		Profile button brings the user to profile page	Profile page loads when clicked	✓
		Logout button sign the user out	Logout button should log the user out and bring them back to the	✓

			login page	
	Visibility of navigation bar	User is logged in	Navigation bar show up	✓
		User is not logged in	Navigation bar does not show up	✓
Dashboard	Able to access the website	User is logged in	Dashboard should be shown	✓
		User is not logged in	Unable to access dashboard and redirected to log in page	✓
	Revenue overview	User is able to see the correct revenue for the following category: total, average, daily	Revenue show is correct, ie daily revenue is 0 when the user have not input any sales yet	✓
		Revenue chart of the user is shown	Only the last 7 days revenue is shown	✓
		Click on the points of the revenue chart will display the revenue count of that day	Correct revenue count for that day is shown	✓
	Transaction history	Transaction history is shown	All the transaction records are shown in decreasing order	✓
		Appropriate information for each transaction is shown in the table	TransactionID, date&time, number of items and revenue is shown for each transaction	✓
	Invoice	Clicking on the receipt got the individual will render the invoice for it	Invoice for the transaction will be shown	✓
		Invoice will show address if it is saved in the profile tab	Address will be shown at the top of the invoice	✓
		Invoice will show contact number if it is saved in the profile tab	Contact number is shown at the top of the invoice	✓
		Invoice will not show the address and	Address and phone number is not shown	✓

		contact field if the info is not available		
		TransactionID, Date and time of the transaction will be shown in the second part of the invoice	Correct information is shown	✔
		Information of items involved in the transaction is shown	Details of items sold is shown (count, title, price etc)	✔
		Total and tax (7%) is accurate	Total and tax(7%) of the transaction is accurate	✔
Cashier	Able to access the website	User is logged in	Cashier should be shown	✔
		User is not logged in	Unable to access Cashier and redirected to login page	✔
	Inventory	There is nothing in inventory	Nothing shows up in inventory tab	✔
		Items present in inventory	Inventory displayed with correct items and quantities	✔
		Inventory not loaded immediately	Loader displayed then inventory displayed once information received	✔
	Cart	Cart stored in local storage	Cart loaded with previously loaded items when page is reloaded/refreshed	✔
		Nothing in cart	Nothing shown in cart tab	✔
		Items added to cart	Cart immediately updates with item information	✔
		Multiple items added	Quantity in cart correctly reflects number of items added	✔
		Different items added	Cart should grow into an infinite scrollable list of items	✔

		Remove from cart	Item removal should be reflected properly in both the cart tab and local storage	✓
	Checkout	Checkout button	Quantity of items in cart should be properly deducted from inventory stock in backend	✓
		Inventory display after checkout	Inventory display should immediately refresh to reflect updated stock count	✓
	Receipt tab	Receipt	Receipt should reflect overall price of items in cart	✓
		Receipt scrolling	Receipt should grow to right before checkout button then become an infinitely scrolling tab	✓
		Receipt for local storage of cart	Receipt should still reflect correct total after refreshing/reloading page	✓
Inventory	Able to access the website	User is logged in	Inventory should be shown	✓
		User is not logged in	Unable to access inventory and redirected to log in page	✓
	Table to show inventory	There is item in the inventory	Details of the item will be shown in the table	✓
		There is no item in the inventory	A line stating that there is no inventory will be shown	✓
	Adding products to inventory (create)	Able to add in new inventory by manually filling up the form	A new product will be added to the inventory and shown	✓
		Clicking add product button or close will close the modal	Modal to add inventory will be closed	✓

		Able to add inventory by using CSV	All the new products in the CSV are uploaded	✓
	Editing product (update)	Product details can be shown after clicking on the edit button	Details are written on the form as a "hint".	✓
		Able to update the item	Clicking save will save the changes to the product	✓
	Deleting product	Clicking on "delete?" button should change it to a confirmation button	Button changes to the confirmation button of "delete" and "cancel"	✓
		Clicking on the confirm delete button will delete the item	Product is removed from the inventory and won't be shown	✓
Sales	Able to access the website	User is logged in	Sales page should be shown	✓
		User is not logged in	Unable to access sales page and redirected to log in page	✓
	Revenue chart	Revenue chart is shown	Revenue chart shows the transaction records for the last 28 days	✓
		Clicking on the points will show the revenue of the specific day	Revenue for that day is shown	✓
		Revenue and profit is shown in the following category: all time, year-to-date, monthly	Revenue and profit is shown	✓
	Profit, revenue, sales chart	Profit chart shows the comparison between the profit in the last 3 months	Profit chart shown is accurate and for the last 3 months	✓
		Revenue chart shows the comparison between the profit in the last 3 months	Revenue chart shown is accurate and for the last 3 months	✓
		Sales chart shows the comparison between the profit in the last 3 months	Sales chart shown is accurate and for the last 3 months	✓