

# Prototype

## Overview

A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from.

I shall prototype the majority of the admin's, staff's & customer's views. This will allow me to prototype the essential part of my solution, the customer creating and placing an order and the staff & admin editing and viewing data about this order and customer.

## Features that I have included:

### General:

- I have prototyped the entire database structure, models and entity relationships in my design chapter. I did this so that I could see if my entity relationship was correct or if I would need to edit it.
- I have created the structure for all of the Django apps that I think I will need. I did this so that I will be able to see if this plan is enough or if I need to break some of my planned apps up, into smaller apps or if there are some problems that would require their own Django apps.
- Account App – I have created almost the entire account app for this app This includes:
  - Creating accounts
  - Login
  - Account email recovery, through password resets
- Dashboard – This is the system's main menu, it will be different based on the user's access level. This is required for the user to navigate through the system

### Customer:

- Cart – This is critical for the customer to create an order
  - Adding to the cart
  - Removing from the cart
  - Updating the cart
- Menu – Critical for the customer to be able to make orders. This will demonstrate both ways the customer will be able to find a product
  - Sort by category
  - Search for a product
- Product Page – Allows the customer to add the product & quantity to the cart, also to view details on the product. This is required so that the customer can make an order.
- Cart Page – Allows the customer to view their current order, continue shopping or go the checkout. This allows the user to make changes to their order before the checkout
- Create order – This allows the customer to save the order.
- They can create an order of multiple order items, each of different individual quantities – required to prove that this system can do the basic functionality required.
- They can view their saved & previous orders
- Search for items – demonstrates how the user would find the product that they want in the system other than simply scrolling through the entire product list.

- The customer can view the pdf of their receipt – this proves that the entire ordering process is contained in my solution, there is no need for any external software to complete the transaction
- The customer will receive an email confirming the order – This shows how my system will notify the customer that the order has been successful.

#### Staff & Manager:

- Can view orders – this proves that the ordering system is fully functioning
- Can sort orders by attributes – Demonstrates how the orders will be organised
- Order can be delete the order here – Required to show that once an order has been created it can be cancelled, as this was not prototyped in the customer's view
- Can update the status for the order – this was prototyped to demonstrate how the system would work in real time as the order was worked on
- View more details on the order – This allows the staff to view the order parts that make up the order. This is required as the cook's view is not prototyped, and this is similar to how the cook would use the system.
- Can update the status for the order part – this was prototyped to demonstrate how the system would work in real time as the order was worked on
- Order can be delete the order part here – Required to show that once an order has been created it can be changed, as this was not prototyped in the customer's view
- Can view, delete & sort customers and view more details on the customer – This was prototyped to show how the staff & managers will be able to view & interact with the customer table. They can also view some data processing on the customer such as how often they order.
- Can view pdf receipts for each order – This shows how my system will allow the staff and admin to look back at previous orders and view their receipts. This is required so that there is no need for paper copies
- Can view multiple pdfs about the customer and/or their order's and/or their order items – This shows how my system will allow the staff and admin to look back at previous orders of customer's and view their receipts. This is required so that there is no need for paper copies

#### Manager:

- Can view, delete, edit & sort the employees – This was prototyped to show how the manager can change the access level of different employees (staff/cook/driver)
- Can view, delete, sort & create categories – This was prototyped as it is critical to how the user sees the products in the menu. It also shows how new categories will be added to the system
- Can view, delete, sort, create & hide products – This was prototyped as it is critical for the customer to have some products to order. It also shows how new products are added.
- Has access to all other access levels, the staff, driver and cook. This is because they should be able to view and use all of their functionality as well, but it would be too impractical and wasteful to reprogram it.

#### Driver:

- Can view and sort the live orders and upcoming orders – shows how the deliver staff will be bale to view what orders & order items that will be delivering and to what addresses.
- They can view a pdf file of each order item that contain information about that order item and the customer so that different order boxes will not get mixed up.

- They can view a pdf file of the receipt so that they can view all of the details of the order and some information on the customer.

#### Features I have not added:

- Validation – All data entered will be test data so there will be no need for validation
- Cook view – this view is not essential for demonstrating how the basics of the system would work or they would be too similar to the staff's view to added separately for the prototype.
- Printing receipts – This is not integral to the ordering system and can be solved by printing the pdf manually
- Ingredients & suppliers – This is irrelevant to the basic ordering system so it is not necessary
- Deliver zones – This would be too complex for this system at this current time and would also be impractical as I would have to create a geo app.
- Target times – non essential for this prototype
- Coupon system – Too complex for this prototype
- CSV exports – Unnecessary to illustrate the main function of this system

Reports – Not integral to the basic ordering system so it shall not be added

## Screenshots:

Log in

Class Not Found

Prototype Chapter

Prototype v1.pdf - Google Drive

127.0.0.1:8000/login/?next=/

Apps Beezer Brorhood YouTube Netflix School Computing Halo WayPoint R3dDlt Club Penguin TV Shows Amazon Games MyDomino's | D

Pedro's Pizza Hut

Login

### Log in

[Register](#)

[Forgotten your password?](#)

Please, use the following form to log in:

Username:

Password:

Log in

This is the home page for users who have not logged in yet. The user can register a new account, retrieve their account if they have forgotten their password or log in. The links will take the user to the appropriate pages.

Below is the page for registering a new account. Once the users registers,

This starts off from the customer's view.

The cook app has been created but not prototyped. The customer, staff and manager's apps have been created and prototyped. The driver's app has been created but not fully prototyped.

- My prototype will clearly show the prototype allowing:
  - A customer to join
  - A customer to place an order for many items
  - A customer receiving their bill through email and pdf
  - The delivery invoice/data that the driver sees
  - The admin view of all orders

Create an account

Please, sign up using the following form:

Gender: ☒

Username:

First name:

Last name:

Email:

Address:

Postal code:

City:

Password:

Repeat Password:

Create my account

Here is the greeting page once the user has registered their account. They are then directed to log in by following the link.

Welcome

Pedro's Pizza Hut

Login

Welcome Conor!

Your account has been successfully created. Now you can [Log in](#).

Save password?

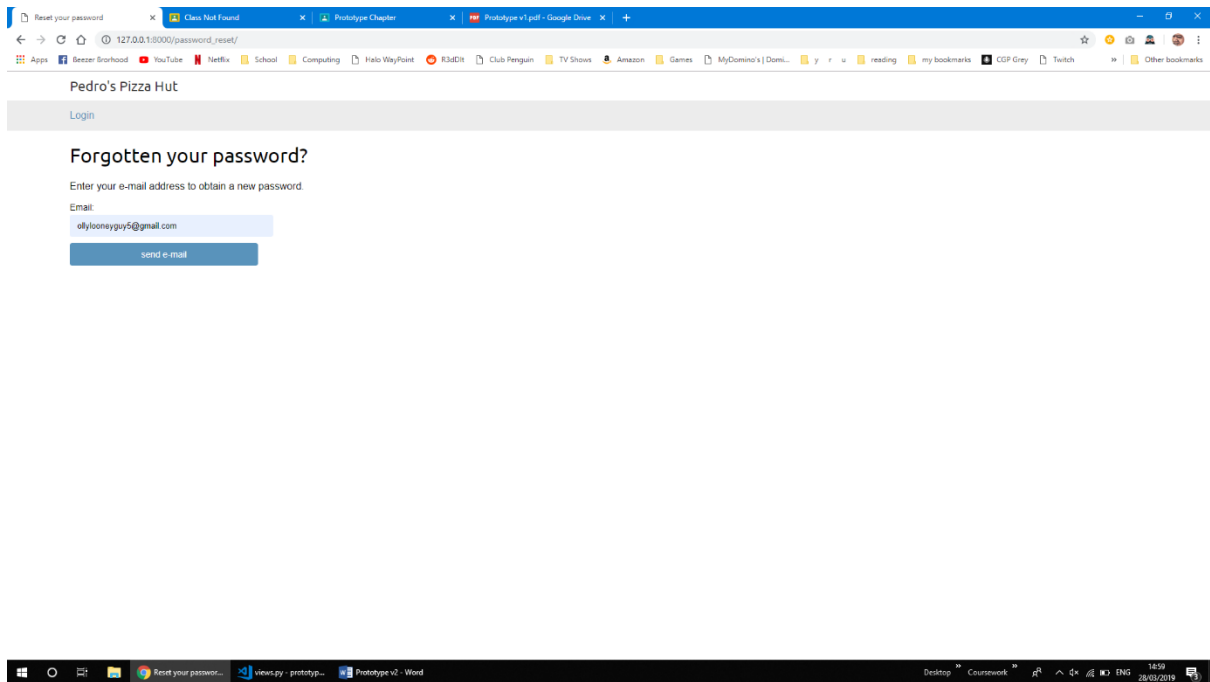
Username: ConorL

Password: \*\*\*\*\*

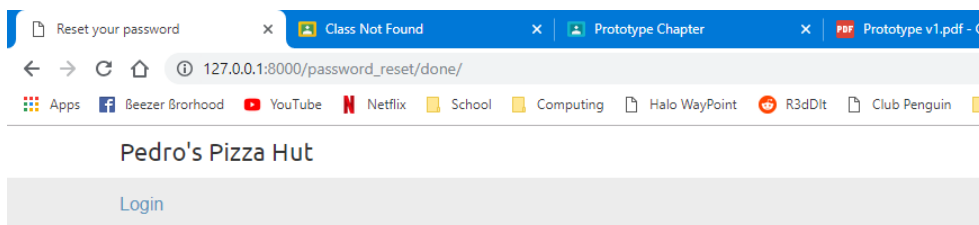
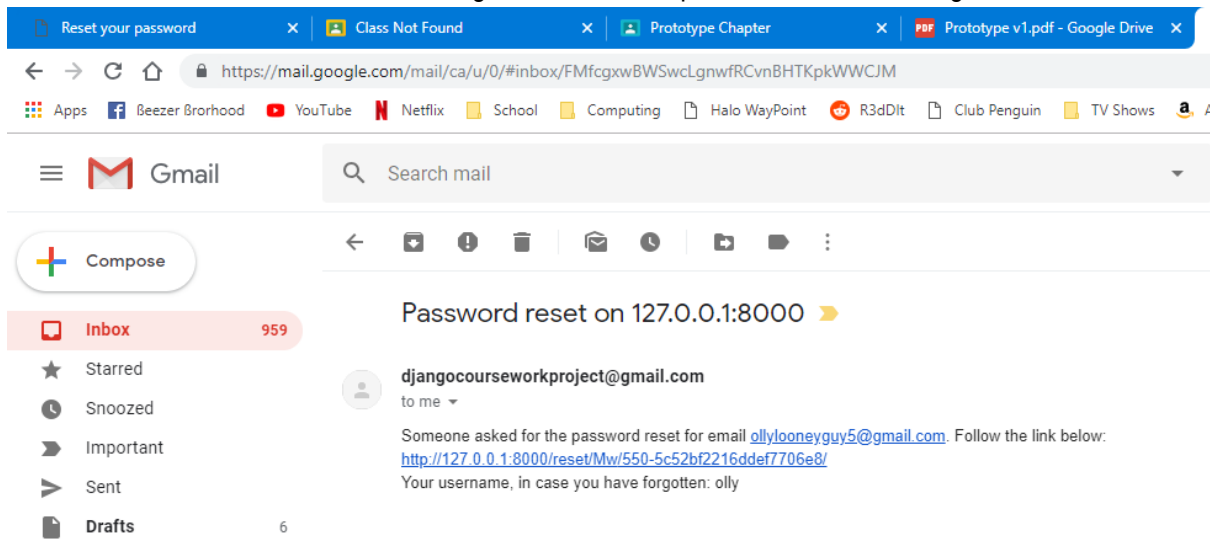
Save Never

Passwords are saved in your Google account so that you can use them on any device

Below is the account retrieving page for when users have forgotten their passwords. The users enter their email into the Django form and then my code shall e- mail them instructions on how to retrieve their account and a specific link for them to do something.



Here is an example of the email that is sent to users who are trying to retrieve their accounts. My code generates a specific link for that user that only lasts for a certain amount of time and once another email is sent to that account for retrieving an account, the previous link will no longer work.



## Reset your password

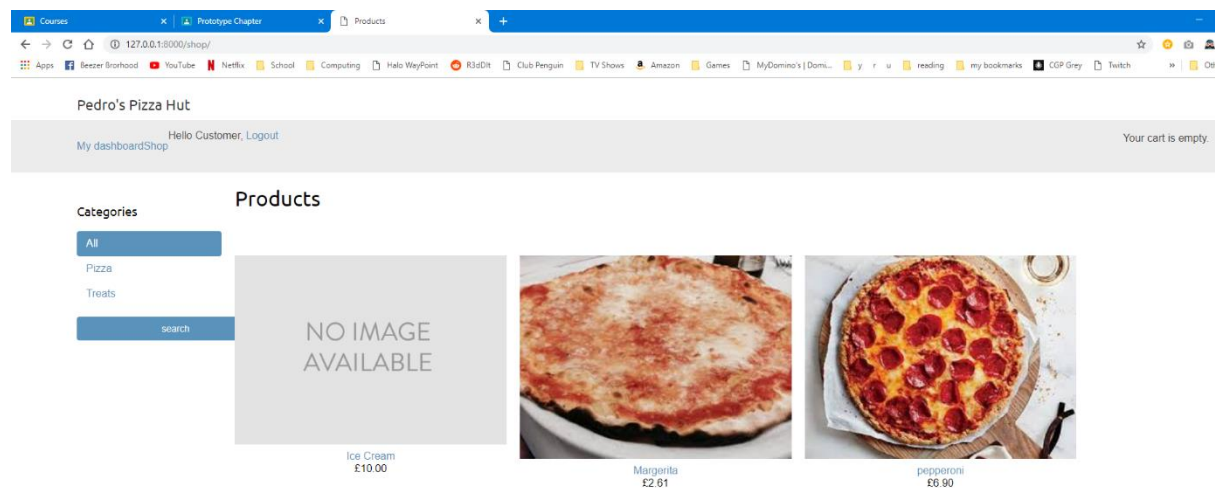
We've emailed you instructions for setting your password.

If you didn't receive an email, please make sure you've entered the address you registered with.

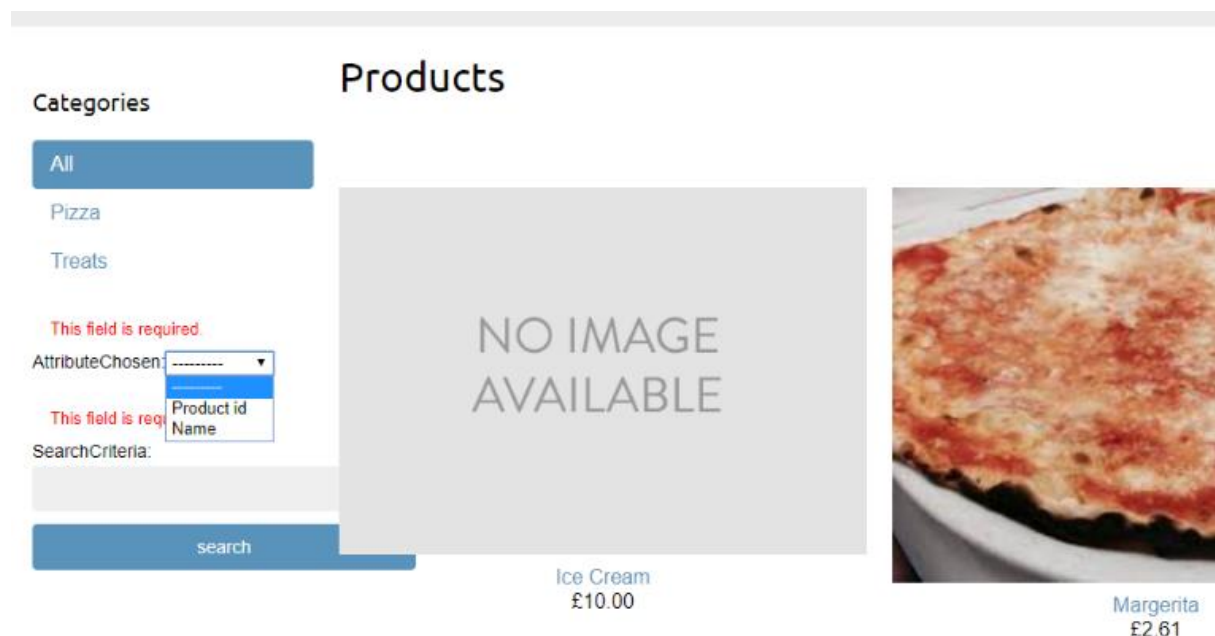


When the user logs in with a customer access level account they are brought to the page above, the dashboard. From the banner in near the top of the page the customer can go to the shop, view the contents of the cart or go the dashboard. All users can logout through the link in the banner.

Inside the dashboard the customers can view a table of their saved orders and all of their previous orders. Inside the table they can save/unsave the order. They can also directly add the item to the cart.



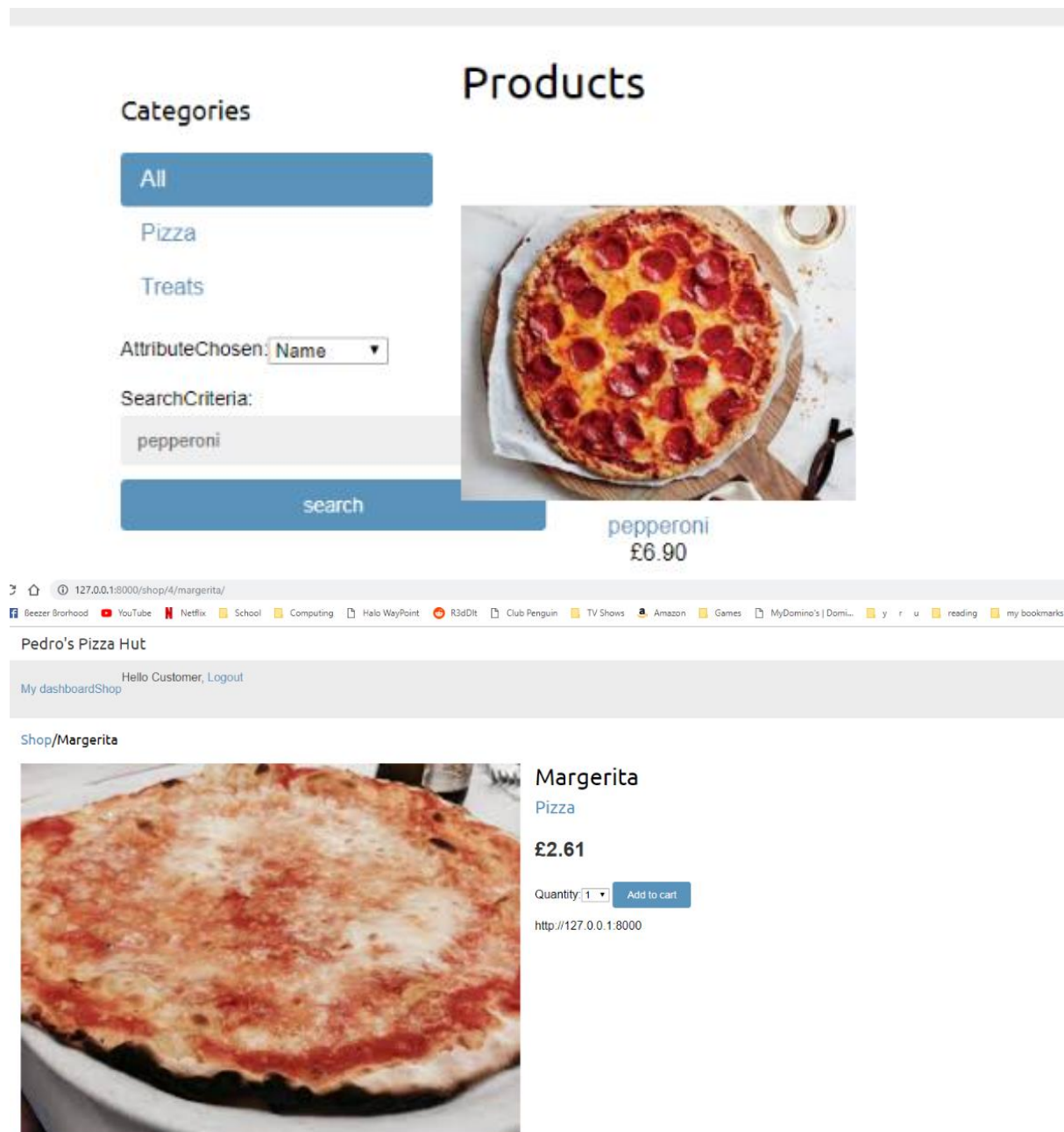
The user can navigate to the shop by clicking the link in the banner. Here is the front of the shop. Here the user can view all of the available products. The user can click on a product to be taken to a new page with product details specifically for that product. In the side bar the user can click on a category and will then be shown a page specifically for that category, with products only in that category. From that page they can still click on a product to go to that product's page. The user can also click on the search button which will open a Django form.



Currently the user can only search by the product ID and name, however in the full system this will be replaced by name and popularity.

As you can see in the next screenshot the search works.

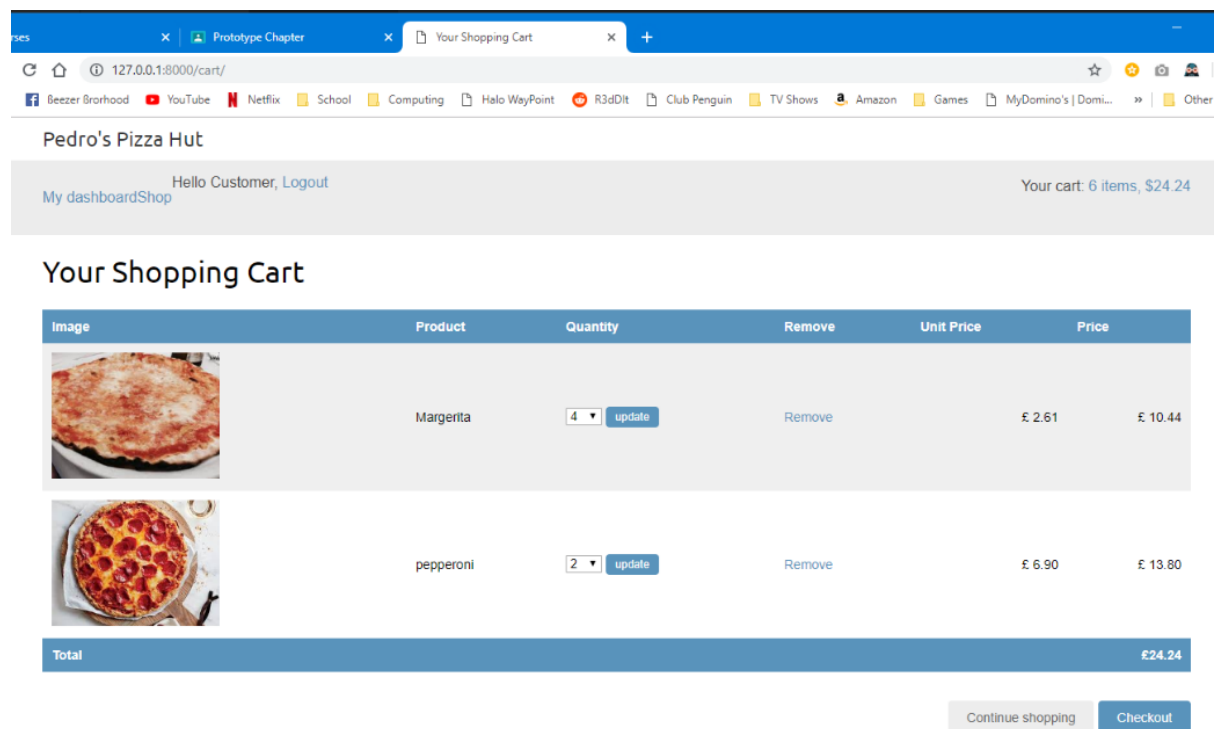




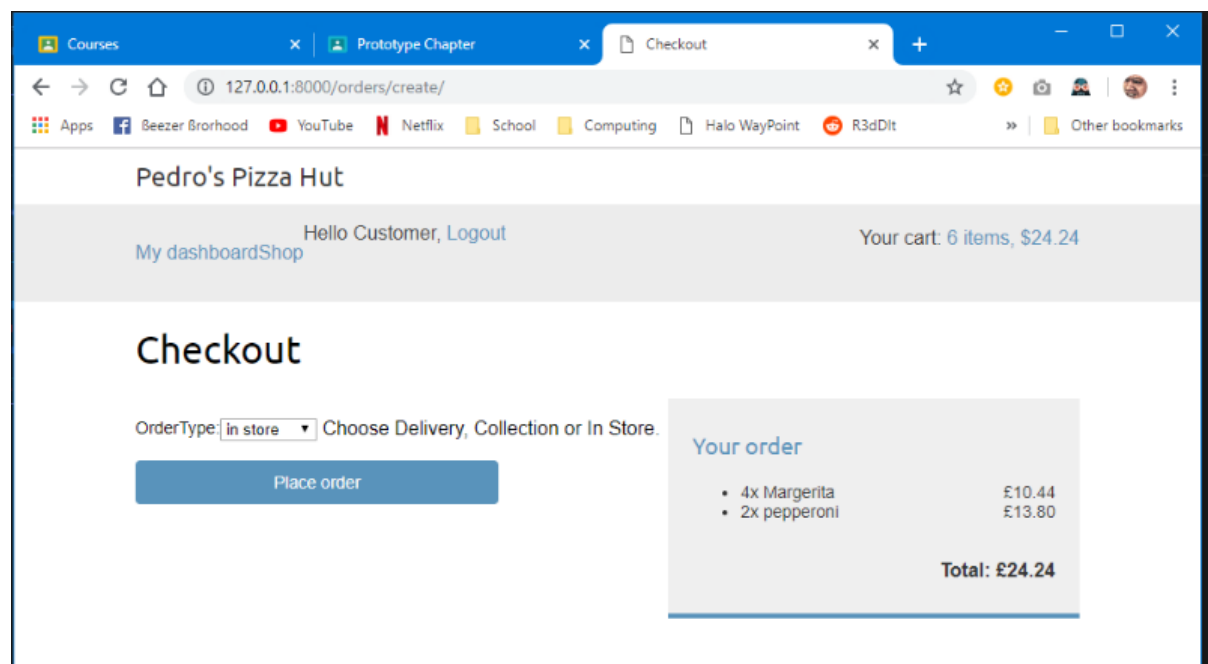
Here is the product detail page. At the top the user is shown where they are by the “Shop/Margerita”, so that they can quickly navigate back. There is also a link to the category page underneath the title of the product. The user can select their chosen quantity from the drop down menu and then add the item(s) to the cart by clicking the “Add to Cart” button. Below that is a description and next to the information about the product is an image of it, that is chosen by the manager.

Below is the screenshot of what happens when the user clicks the “Add to Cart” button. The user is brought to a different page where they can view their cart. In the cart they can see all of the items that they have added and some details about them, such as: Image, Product, Quantity, Unit Price and Price of the item(s). The user can then view the total as the bottom. The user can remove an item from the cart by clicking the remove link. The user can also edit the quantity by selecting a different value from the drop down menu and clicking the update button. The user can click on the “Continue shopping” button to go back to the shop page and order more items. They can also click on the “Checkout” button to go to the checkout. In the top right of the banner, the user’s current cart contents are shown.

I have added some more items to the cart to show that the user can order more than one item at a time.



This proves that in my prototype a user can create an order of multiple parts.



This is the Checkout page for the user. The user only has to input what type it is, i.e.: delivery, collection or in store. This is because the system automatically fetches the user's id and uses that to create a relationship between the customer table and order table. Also, all of the data in the cart is automatically saved.

The user chooses the order type by selecting an option from the drop down menu. The contents of the order are shown at the side. Once the user clicks "Place order" the order\_create view function is called which actually creates the order.

```

def order_create(request):
    cart = Cart(request)
    if request.method == 'POST':
        form = OrderCreateForm(request.POST)
        if form.is_valid():
            cd = form.cleaned_data#old code in book did not work with saving
the customer id too
            order = Order.objects.create(customer_id = request.user.id,
                                         orderType = cd['orderType'],)
            order.orderslug = order.id
            order.save()

            for item in cart:
                OrderItem.objects.create(order = order,
                                         product = item['product'],
                                         price = item['price'],
                                         quantity = item['quantity'])

            #clear the cart
            cart.clear()
            items = OrderItem.objects.filter(order_id = order.id)
            customer = MyUser.objects.get(id = order.customer_id)

            orderpartmsg = ""
            for item in items:
                orderpartmsg = orderpartmsg + str(item.quantity) + " " +
str(item.product.name) + ": " + "£" + str(item.get_cost()) + "\n"

            ordertype = order.get_type()
            from_email = settings.EMAIL_HOST_USER
            send_mail(
                'Receipt',#subject
                ("Dear, " + str(customer.first_name) + " here is your order
receipt:\n"
                 + "Order ID: " + str(order.id) + "\n"
                 + "Created at: " + str(order.id) + "\n"
                 + "Order Type: " + str(ordertype) + "\n\n\n\n\n"
                 + orderpartmsg),#message
                from_email,
                ['ollylooneyguy5@gmail.com'],#email to be sent to
                fail_silently=True,
            )
            return render(request,
                          'orders/order/created.html',
                          {'order': order,
                           'items': items,
                           'customer': customer})

```

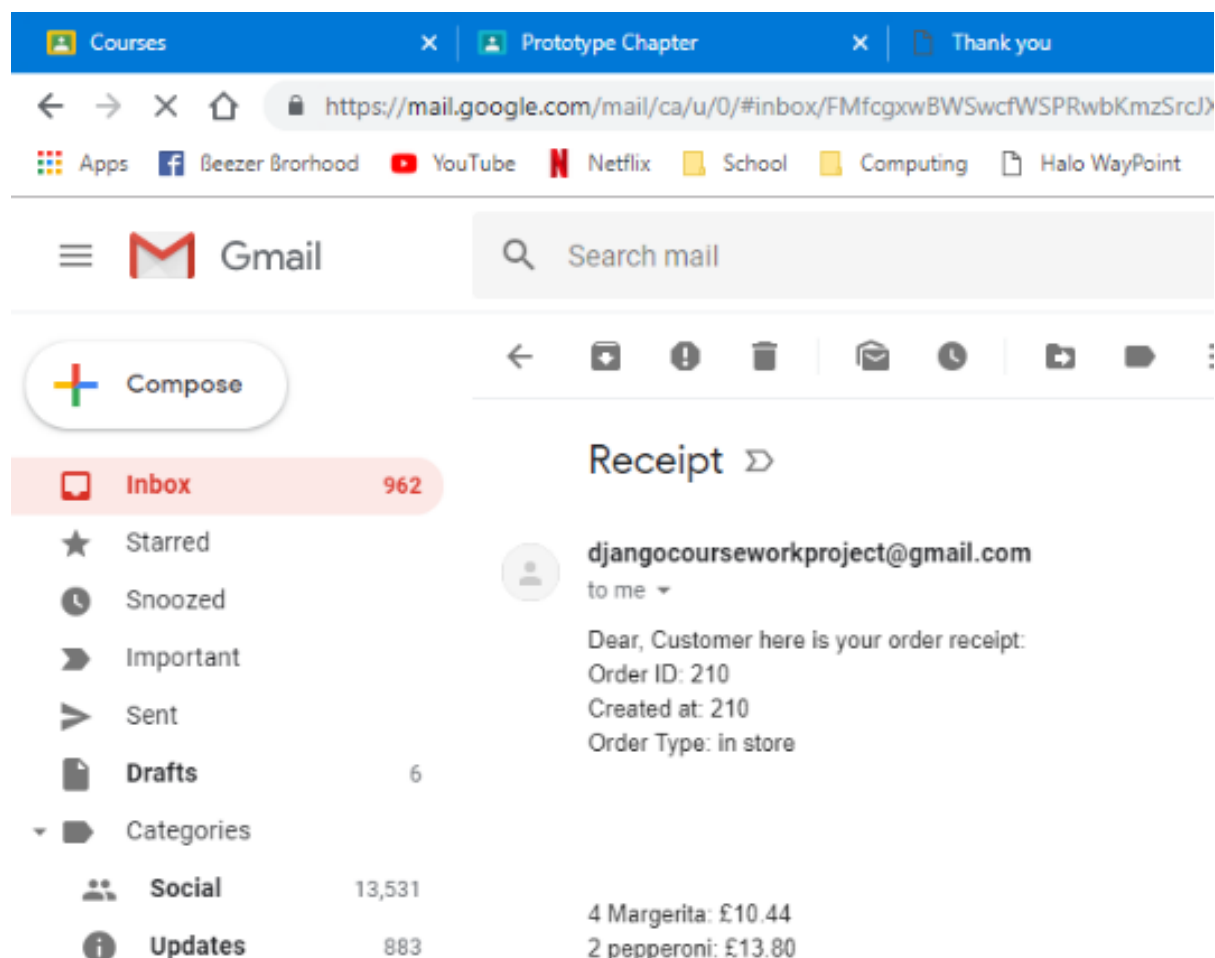
```
#lines 24-30 were indented too much causing an error, no http response was
returned, none was returned
else:
    form = OrderCreateForm()

    return render(request,
                  'orders/order/create.html',
                  {'cart': cart,
                  'form': form})
```

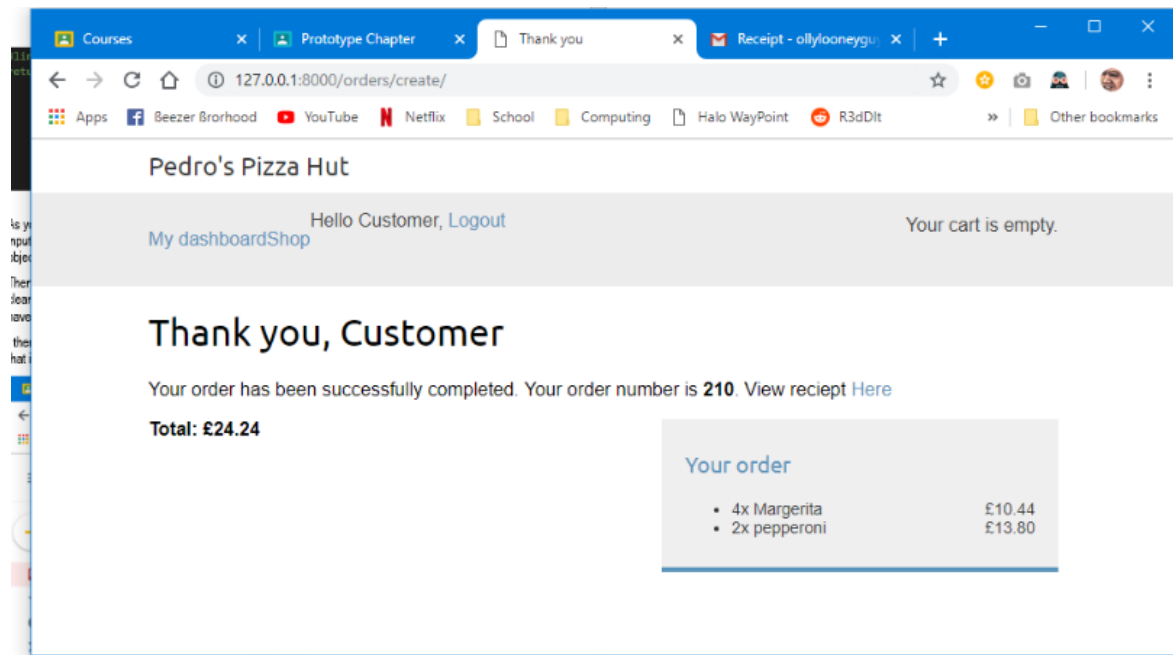
As you can see in the code, an order object is created with the order type and customer id being inputted first, the order id is auto increment, Then the orderslug is created and then I save the order object to the database, creating the order.

Then I loop through the items in the cart creating an OrderItem object for each item in the cart, then I clear the cart from the user's browser's data so that next time the user visits the site their cart will not have their previous order still in it.

I then send the user an email receipt before directing them to the order created page to let them know that it was a success.

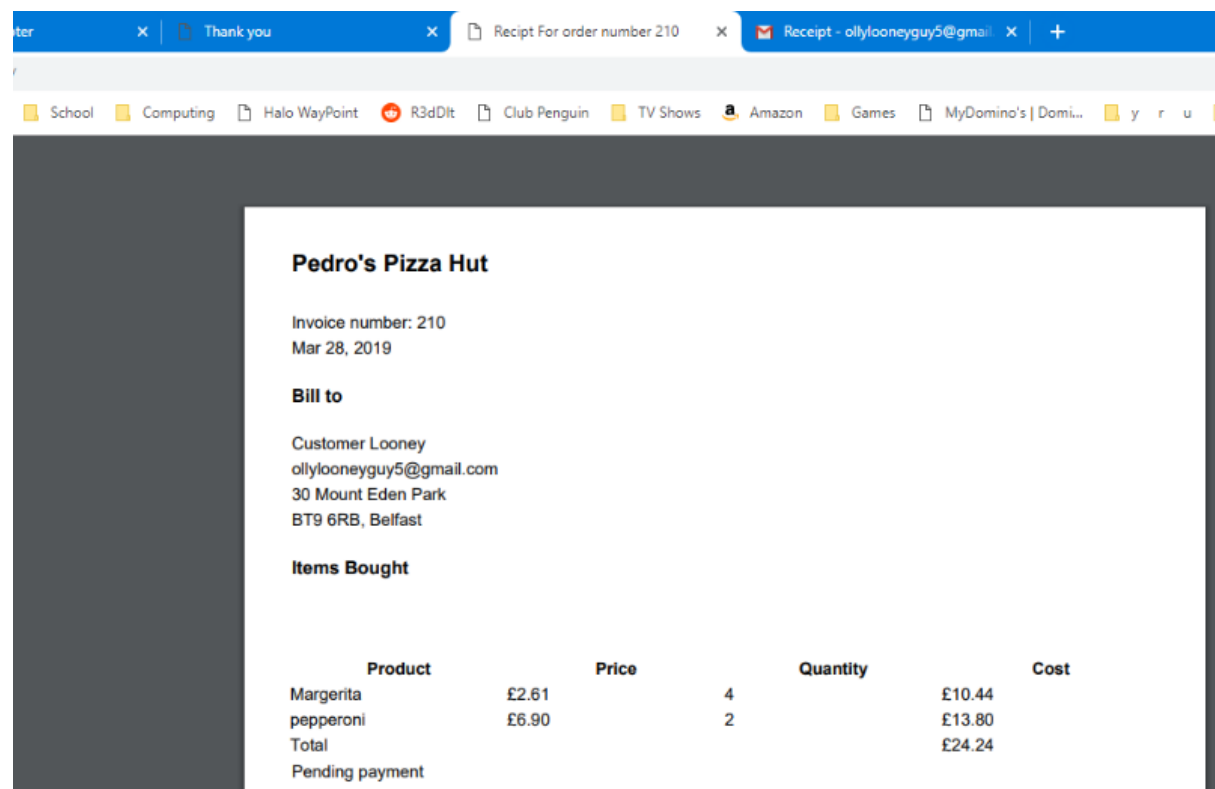


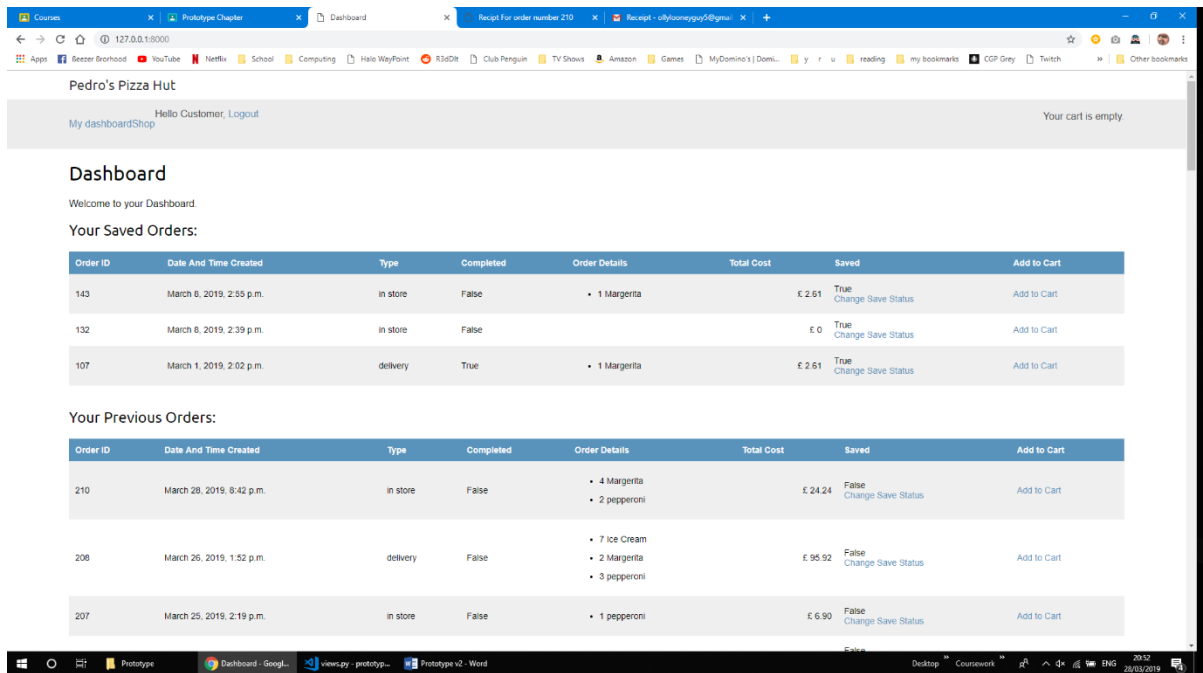
Here is the email generated and sent to confirm the order.



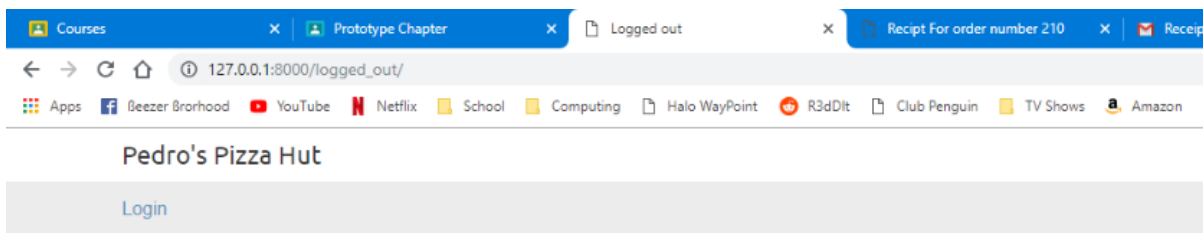
Here is the order created page that lets the user know that the order has been a success. In the full system I shall give the user a link to a page where they can track the progress of their order, and if it is a delivery the progress of the driver.

The customer can view all relative information, the cost and order id, and the order itself is in the sidebar. The reason it says, "Thank you, Customer" is because for the purposes of this test, the account's name is "Customer". The user can also click on the link to view a pdf file of their receipt of their order in another tab. Below is the receipt of the previous order. The receipt displays all the relative information.





The user can then go navigate back to their dashboard and see that the order has been added to their previous orders and that their cart has been cleared, so they can create another order.

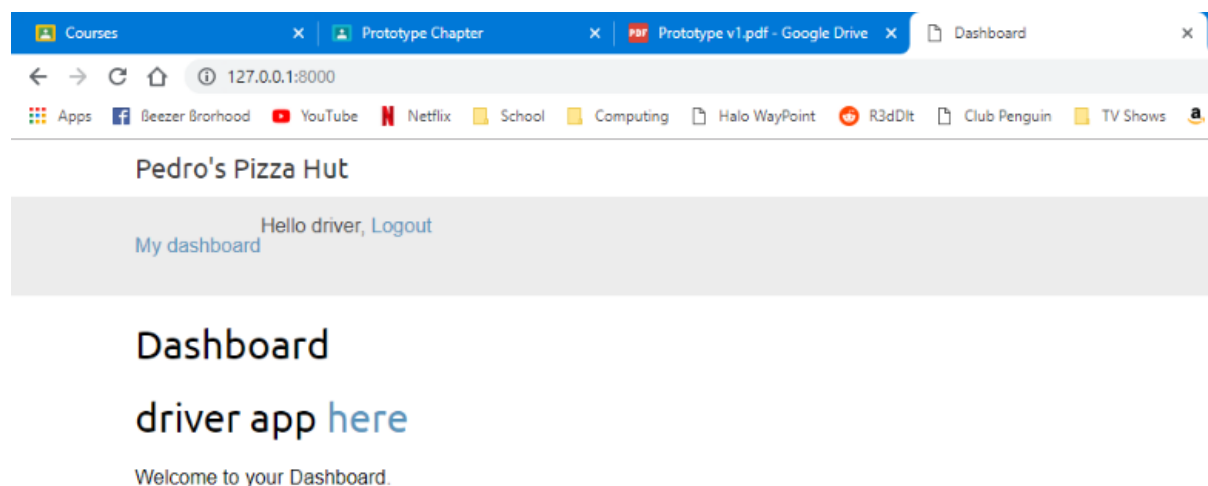


## Logged out

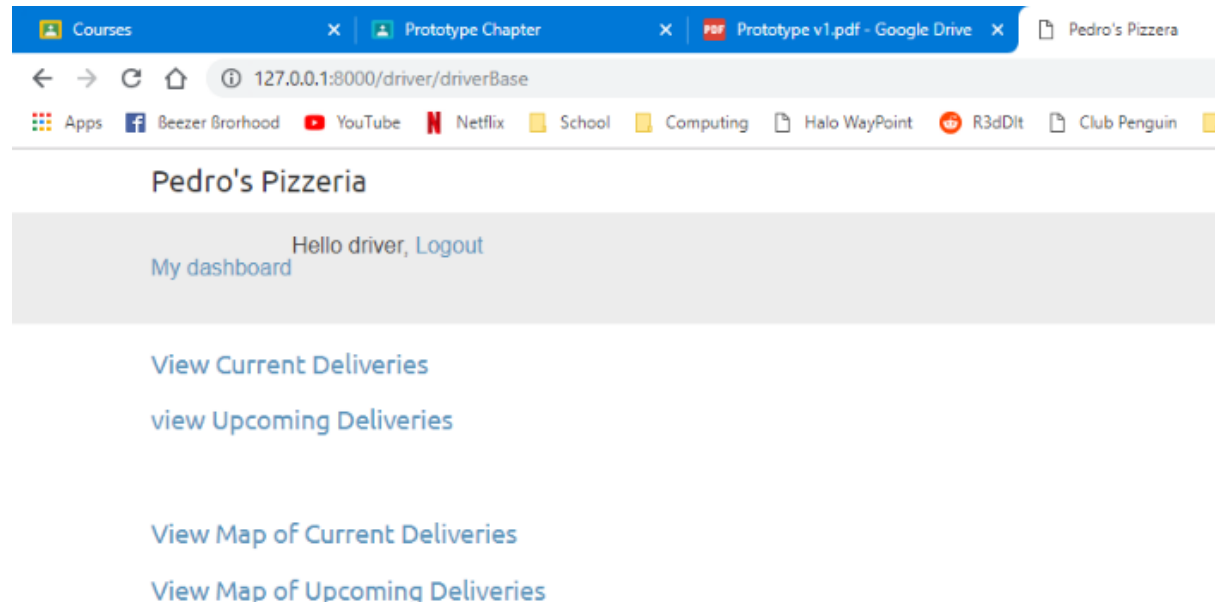
You have been successfully logged out. You can [Log in again](#).

Then the customer can logout like so. That is all of the customer's view prototyped.

Here is the driver's view prototyped:



Once an account with a driver access level logs in this is their dashboard. In the full system they will go straight to the driver's app.



Here is the driver's main menu. They can view the live deliveries or the upcoming delivers. What makes a delivery live is if the order's completed attribute has been set to True. In the full version the driver will be able to view these deliveries on maps but not in the prototype.

Below you can see the current deliveries view. I changed the order I made earlier to a delivery in the database for demonstration purposes. All of the relevant information for the driver is displayed here, customer name and address, order id and time, and the cost. The driver can also view the order details page and the pdf receipt of the order.

The pdf receipt is shown underneath. The order detail page will be explained later in the staff view.

127.0.0.1:8000/driver/viewcurrentDeliveries

Beezer Brorhood YouTube Netflix School Computing Halo WayPoint R3dDlt Club Penguin

## Pedro's Pizzeria

Hello driver, [Logout](#)

[My dashboard](#) [Driver Menu](#)

Welcome

Order ID	Customer ID	Customer Name	Customer Address	Date And Time Created	Order Details	Reciept	Payment type	Paid	Total Cost
210	5	Customer Looney	30 Mount Eden Park	March 28, 2019, 8:42 p.m.	<a href="#">View more details on this order</a>	<a href="#">View Reciept</a> <a href="#">Print Reciept</a>	Cash	False	£ 24.24
208	5	Customer Looney	30 Mount Eden Park	March 26, 2019, 1:52 p.m.	<a href="#">View more details on this order</a>	<a href="#">View Reciept</a> <a href="#">Print Reciept</a>	Cash	False	£ 95.92

Prototype v1.pdf - Google L... localhost Receipt For order number 21 Pedro's Pizzeria Pedro's Pizzeria

t:order.customer\_id>/

Computing Halo WayPoint R3dDlt Club Penguin TV Shows Amazon Games MyDomino's | Doml... y r u reading

1 / 1

### Pedro's Pizza Hut

Invoice number: 210  
Mar 28, 2019

**Bill to**

Customer Looney  
ollylooneyguy5@gmail.com  
30 Mount Eden Park  
BT9 6RB, Belfast

**Items Bought**

Product	Price	Quantity	Cost
Margerita	£2.61	4	£10.44
pepperoni	£6.90	2	£13.80
Total			£24.24
Pending payment			



Apps Beezer Brorhood YouTube Netflix School Computing Halo WayPoint R3dDit Club Penguin TV Shows » Other bookmarks

## Pedro's Pizzeria

Hello driver, [Logout](#)

[My dashboard](#) Driver Menu

Welcome

Order ID	Customer ID	Customer Name	Customer Address	Date And Time Created	Order Details	Receipt	Payment type	Paid	Total Cost
193	5	Customer Looney	30 Mount Eden Park	March 23, 2019, 6:20 p.m.	<a href="#">View more details on this order</a>	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>	Cash	False	£ 144.01
184	5	Customer Looney	30 Mount Eden Park	March 22, 2019, 3:08 p.m.	<a href="#">View more details on this order</a>	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>	Cash	False	£ 0
183	5	Customer Looney	30 Mount Eden Park	March 22, 2019, 3:05 p.m.	<a href="#">View more details on this order</a>	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>	Cash	False	£ 138.04

Here is the upcoming deliveries page. It has all the same functionality and data as the live delivery page so I shall not reexplain it.

Apps Beezer Brorhood YouTube Netflix School Computing Halo WayPoint R3dDit

## Pedro's Pizza Hut

Hello staff, [Logout](#)

[My dashboard](#)

# Dashboard

staff app [here](#)

Welcome to your Dashboard.

Once an account with a staff access level logs in this is their dashboard. In the full system they will go straight to the staff's app. Below is the staff's main menu.

Apps Beezer Brorhood YouTube Netflix School Computing Halo WayPoint R3dDit

## Pedro's Pizzeria

Hello staff, [Logout](#)

[My dashboard](#)

[View and Edit Orders](#)

[View and Edit Customers](#)

**Pedro's Pizzeria**

Hello staff, Logout

[My dashboard](#) [Staff Menu](#)

Welcome

[update](#)

Order ID	Customer ID	Date And Time Created	Type	Status	Order Details	Total Cost	Receipt
210	5	March 28, 2019, 8:42 p.m.	delivery	preparing <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 24.24	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>
209	9	March 26, 2019, 10:21 p.m.	in store	preparing <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 10.00	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>
208	5	March 26, 2019, 1:52 p.m.	delivery	preparing <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 95.92	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>
207	5	March 25, 2019, 2:19 p.m.	in store	preparing	<a href="#">View more details on this order</a>	£ 6.90	<a href="#">View Receipt</a>

When the staff clicks on the “View and Edit Orders” link they are brought to this page. Here they can see all of the orders and some details on them. The staff can click on “View more details on this order” to go to another page that will show some more details on the order and the order items. The staff can change the status of the order by clicking “Update Status”. The staff can open the receipt of the order pdf in a new tab by clicking on the “View receipt” link. If the staff clicks on the “Update” button they will be able to order the orders by certain attributes in ascending or descending order.

**Pedro's Pizzeria**

Hello staff, Logout

[My dashboard](#)

Welcome

• This field is required.  
 AttributeChosen:

• This field is required.  
 OrderOfOrders:

[update](#)

Order ID	Customer ID	Date And Time Created	Type	Status	Order
				preparing	<a href="#">View</a>

Here you can see the form, and in the next screenshot you can see the orders have been reordered.

Pedro's Pizzeria

Hello staff, Logout

My dashboard

Welcome

Attribute Chosen: Date & Time Created

Order Of Orders: Ascending

update

Order ID	Customer ID	Date And Time Created	Type	Status	Order Details	Total Cost	Receipt
107	5	March 1, 2019, 2:02 p.m.	delivery	ready <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 2.61	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>
108	1	March 1, 2019, 2:10 p.m.	collection	ready <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 10.44	<a href="#">View Receipt</a> <a href="#">Print Receipt</a>
109	2	March 1, 2019, 2:11 p.m.	delivery	ready <a href="#">Update Status</a>	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 6.90	<a href="#">View Receipt</a>

Here is the pdf receipt the staff can access.

Prototype v1.pdf - Google Drive

Pedro's Pizzeria

Receipt For order number 210

omer\_id>/

uting

Halo WayPoint

R3dDit

Club Penguin

TV Shows

Amazon

Games

MyDomino's | Domi...

y r u

reading

1 / 1

**Pedro's Pizza Hut**

Invoice number: 210  
Mar 28, 2019

**Bill to**

Customer Looney  
ollylooneyguy5@gmail.com  
30 Mount Eden Park  
BT9 6RB, Belfast

**Items Bought**

Product	Price	Quantity	Cost
Margerita	£2.61	4	£10.44
pepperoni	£6.90	2	£13.80
Total			£24.24
Pending payment			

The staff can also delete the order by clicking on the "Delete this order" link.

The next screenshot is of the order detail page.

This shows the staff some basic details of the order, just as it is a prototype, and then the order items and details of the order items.

The staff can update the status, remove or view the individual ticket of that order item. The individual ticket is required for deliveries or collections were the order is broken up into different boxes.

Pedro's Pizzeria

Hello staff, Logout

Staff Menu

Username: Customer


Order ID: 210

Order Completed: True

Order Paid: False

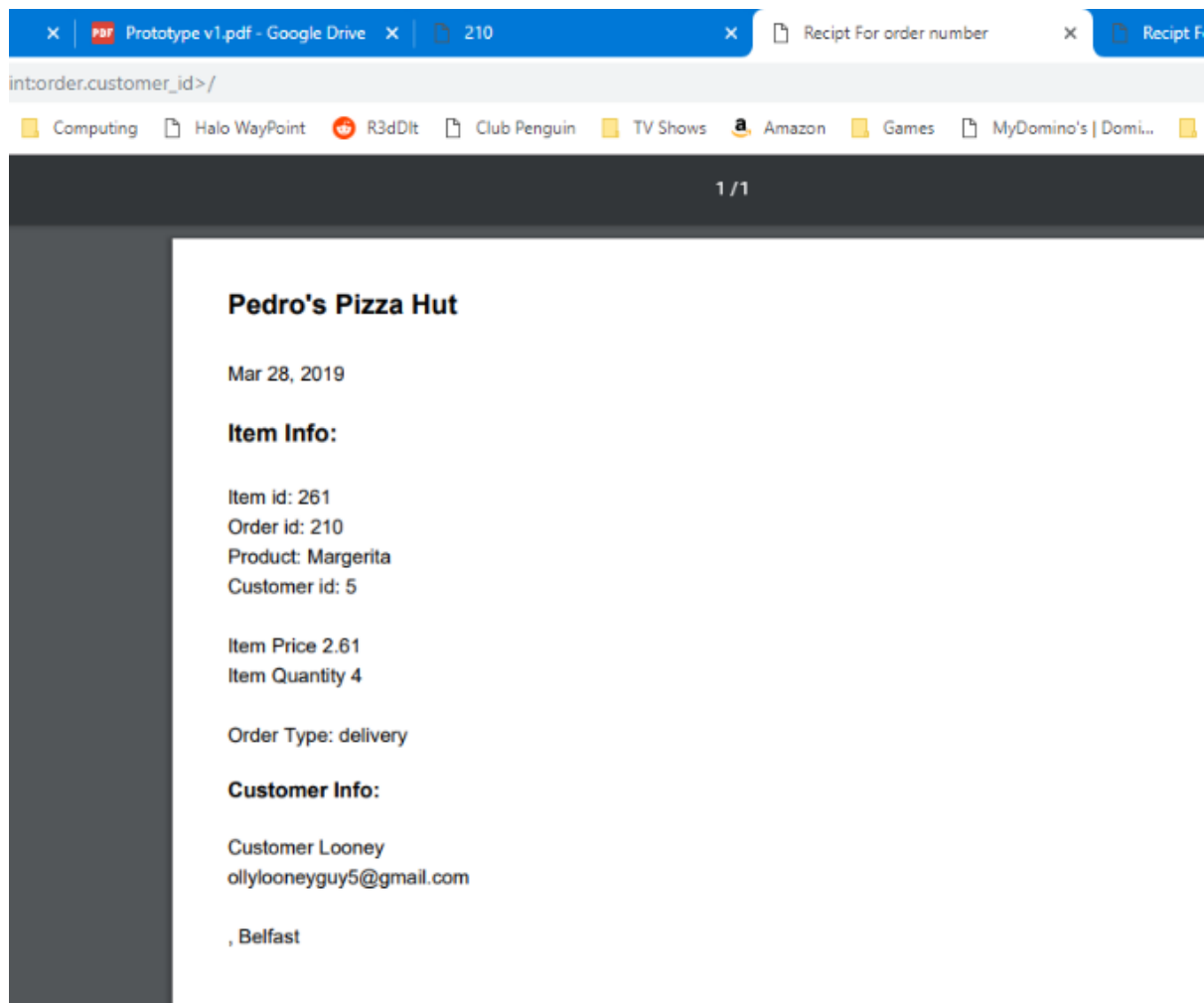
Remove

# of parts: 2

Image	Product	Status	Quantity	Remove	Unit Price	Price	Order Part ticket
	preparing	Margherita	4	Remove	£ 2.61	£ 10.44	View Ticket Print Ticket
	Update Status						

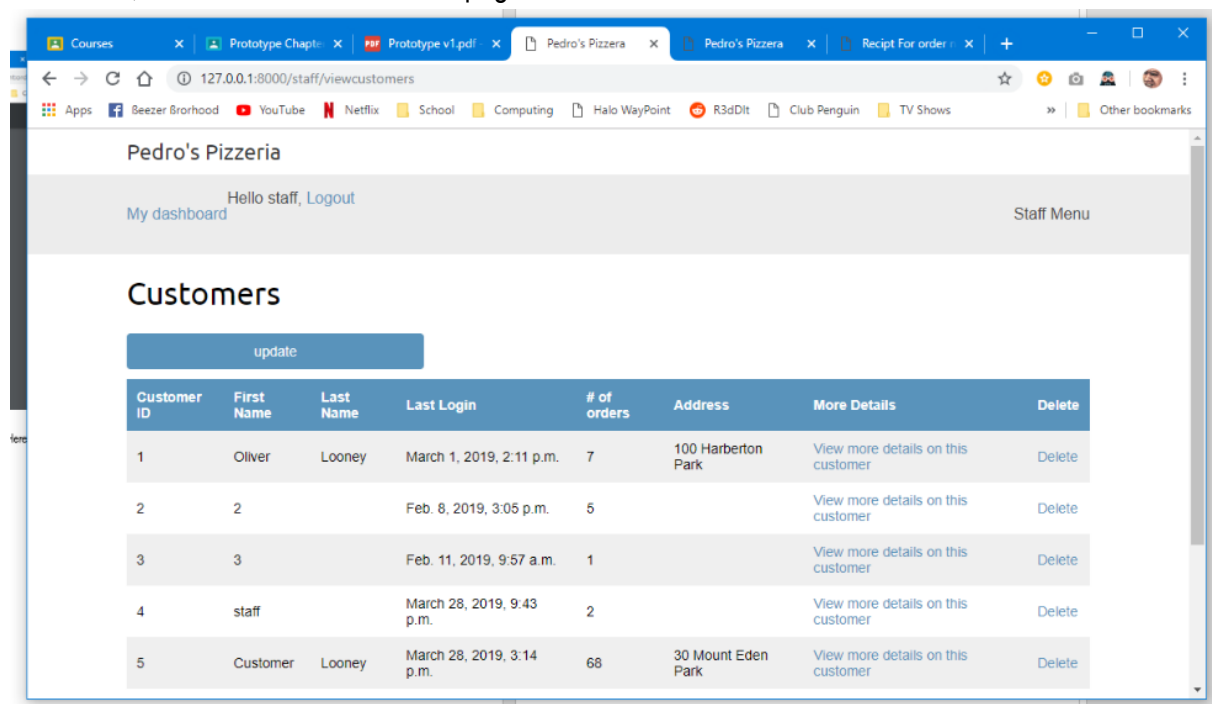
	preparing	pepperoni	2	Remove	£ 6.90	£ 13.80	View Ticket Print Ticket
	Update Status						

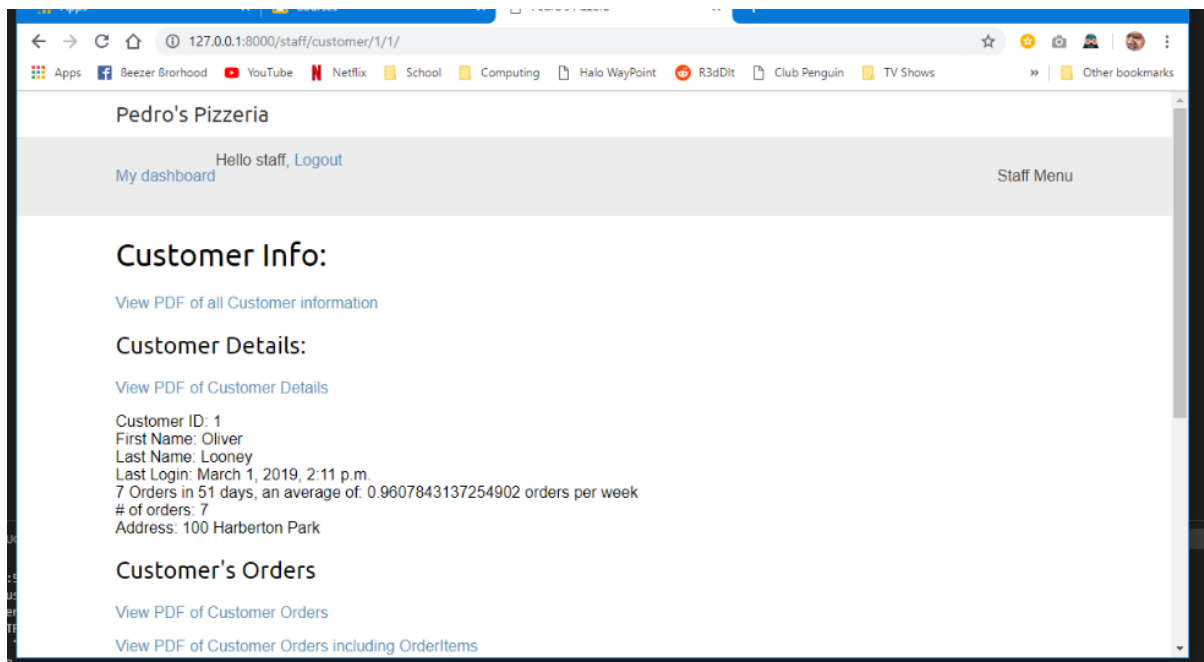
Total £24.24



Here is the order ticket pdf for the margherita in the previous screenshot.

Below is the staff's view customer page. The staff can view information on the customer, reorder the customers, delete customers or view a page of more detail.





# of orders: 7  
 Address: 100 Harberton Park

### Customer's Orders

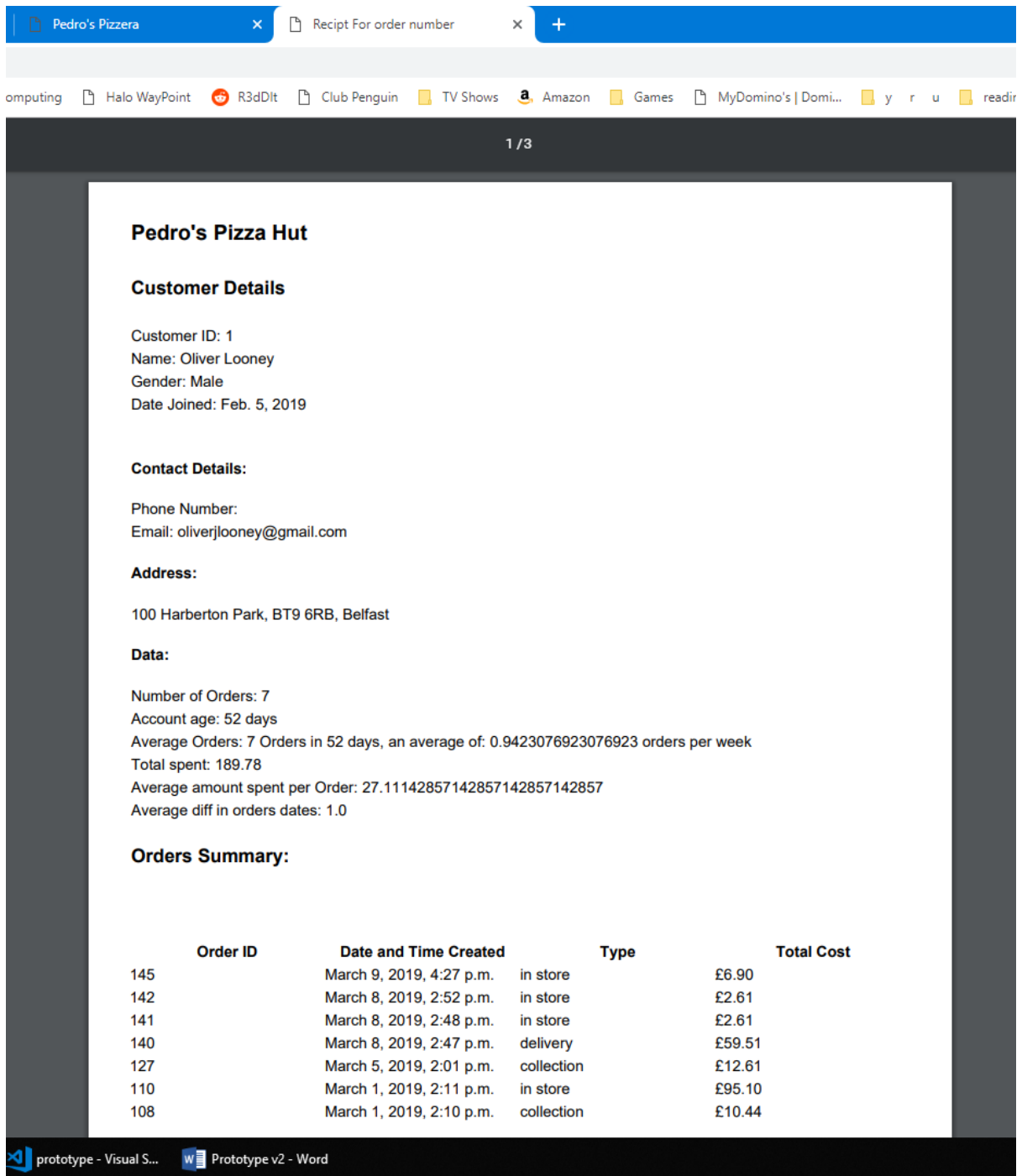
[View PDF of Customer Orders](#)  
[View PDF of Customer Orders including OrderItems](#)

Order ID	Customer ID	Date And Time Created	Type	Status	Order Details	Total Cost
145	1	March 9, 2019, 4:27 p.m.	in store	preparing	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 6.90
142	1	March 8, 2019, 2:52 p.m.	in store	preparing	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 2.61
141	1	March 8, 2019, 2:48 p.m.	in store	preparing	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 2.61
140	1	March 8, 2019, 2:47 p.m.	delivery	preparing	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 59.51
127	1	March 5, 2019, 2:01 p.m.	collection	baking	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 12.61
110	1	March 1, 2019, 2:11 p.m.	in store	ready	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 95.10
108	1	March 1, 2019, 2:10 p.m.	collection	ready	<a href="#">View more details on this order</a> <a href="#">Delete This Order</a>	£ 10.44

Here is the customer detail page. The staff is shown some details about the customer, and then a table of all of the customer's orders. The staff can also view multiple different pdfs of customer data. From the orders table to the staff can go to the view more details page for each order, which has already been shown.

Below I shall show the different PDFs.

First is the view all customer information.



edro's Pizzera

Receipt For order number

+

Halo WayPointR3dDitClub PenguinTV ShowsAmazonGamesMyDomino's | Domi...y r u readin

Orders Summary:

Order ID	Date and Time Created	Type	Total Cost
145	March 9, 2019, 4:27 p.m.	in store	£6.90
142	March 8, 2019, 2:52 p.m.	in store	£2.61
141	March 8, 2019, 2:48 p.m.	in store	£2.61
140	March 8, 2019, 2:47 p.m.	delivery	£59.51
127	March 5, 2019, 2:01 p.m.	collection	£12.61
110	March 1, 2019, 2:11 p.m.	in store	£95.10
108	March 1, 2019, 2:10 p.m.	collection	£10.44

Full Orders and OrderItems:

- Order ID: 145
- Date and Time Created: March 9, 2019, 4:27 p.m.
- Type: in store
- Total Cost: £6.90

OrderItem ID	Product	Quantity	Cost
145	3 - pepperoni	1	£6.90

- Order ID: 142
- Date and Time Created: March 8, 2019, 2:52 p.m.
- Type: in store
- Total Cost: £2.61

OrderItem ID	Product	Quantity	Cost
142	4 - Margerita	1	£2.61

- Order ID: 141
- Date and Time Created: March 8, 2019, 2:48 p.m.
- Type: in store
- Total Cost: £2.61

prototype - Visual S...Prototype v2 - Word



Here is the pdf of customer details.

Pedro's Pizzeria

Receipt For order number

Receipt For order number

ComputingHalo WayPointR3dDltClub PenguinTV ShowsAmazonGamesMyDomino's | Domi...y r ureading

1 / 1

Pedro's Pizza Hut

Customer ID: 1

Name: Oliver Looney

Gender: Male

Date Joined: Feb. 5, 2019

Contact Details:

Phone Number:

Email: oliverjlooney@gmail.com

Address:

100 Harberton Park, BT9 6RB, Belfast

Data:

Number of Orders: 7

Account age: 52 days

Average Orders: 7 Orders in 52 days, an average of: 0.9423076923076923 orders per week

Total spent: 189.78

Average amount spent per Order: 27.11142857142857142857142857

Average diff in orders dates: 1.0

Here is the pdf of the customer's previous orders.

Pedro's Pizzeria

Receipt For order number

Receipt For order number

Receipt For order number

ComputingHalo WayPointR3dDltClub PenguinTV ShowsAmazonGamesMyDomino's | Domi...y r urei

1 / 1

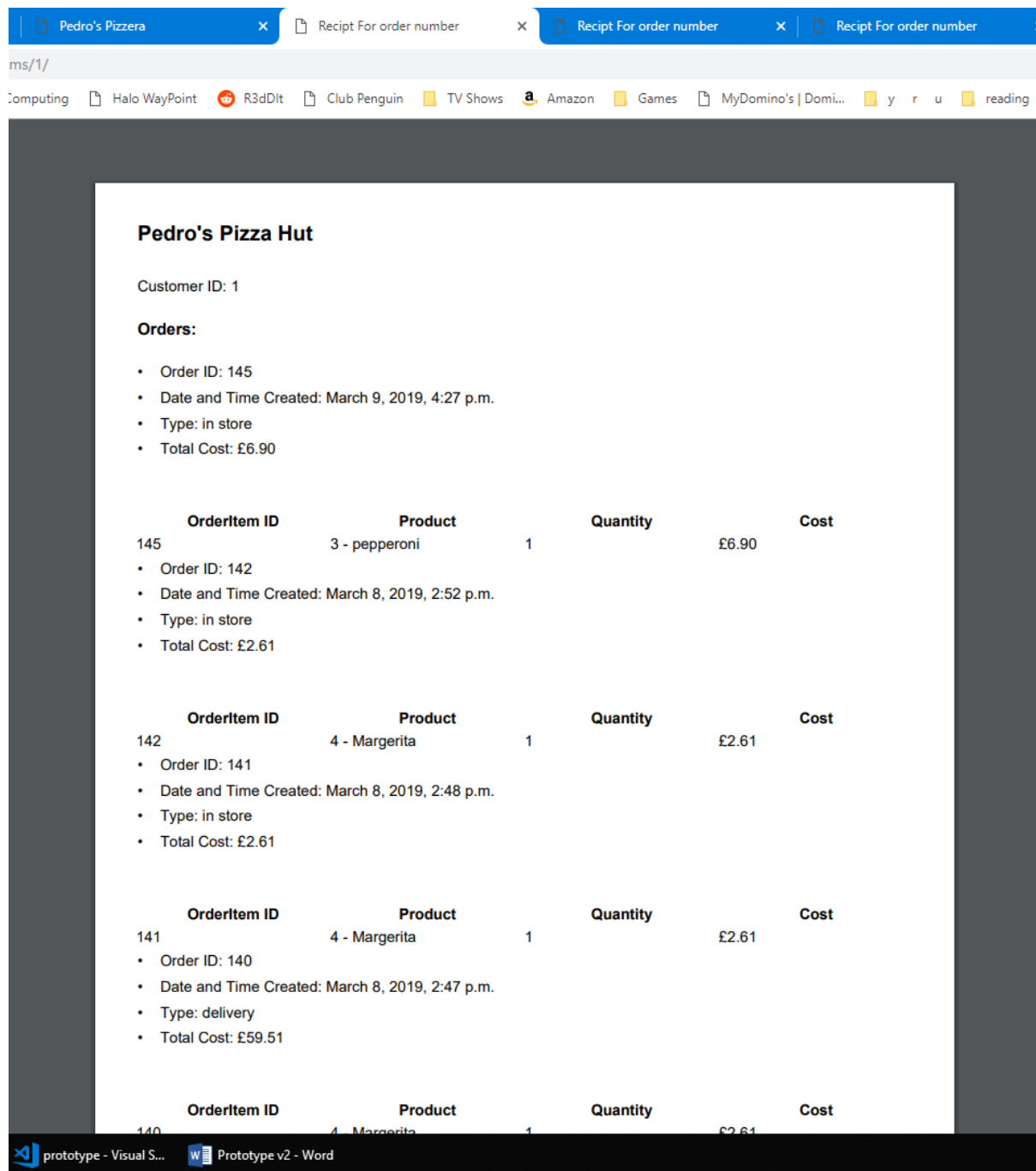
Pedro's Pizza Hut

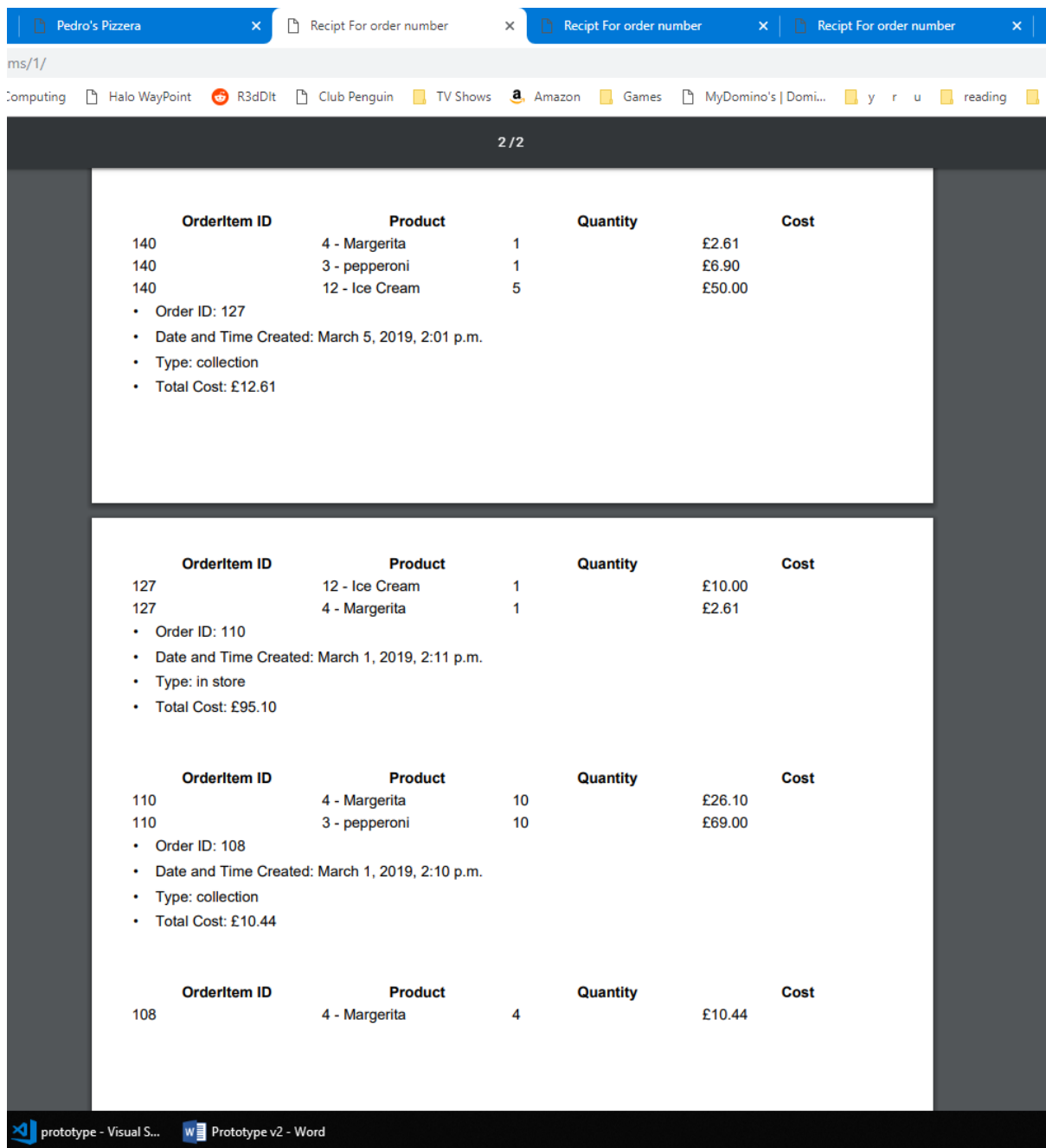
Customer ID: 1

Orders:

Order ID	Date and Time Created	Type	Total Cost
145	March 9, 2019, 4:27 p.m.	in store	£6.90
142	March 8, 2019, 2:52 p.m.	in store	£2.61
141	March 8, 2019, 2:48 p.m.	in store	£2.61
140	March 8, 2019, 2:47 p.m.	delivery	£59.51
127	March 5, 2019, 2:01 p.m.	collection	£12.61
110	March 1, 2019, 2:11 p.m.	in store	£95.10
108	March 1, 2019, 2:10 p.m.	collection	£10.44

Here is the pdf of the customer's order with the order items.



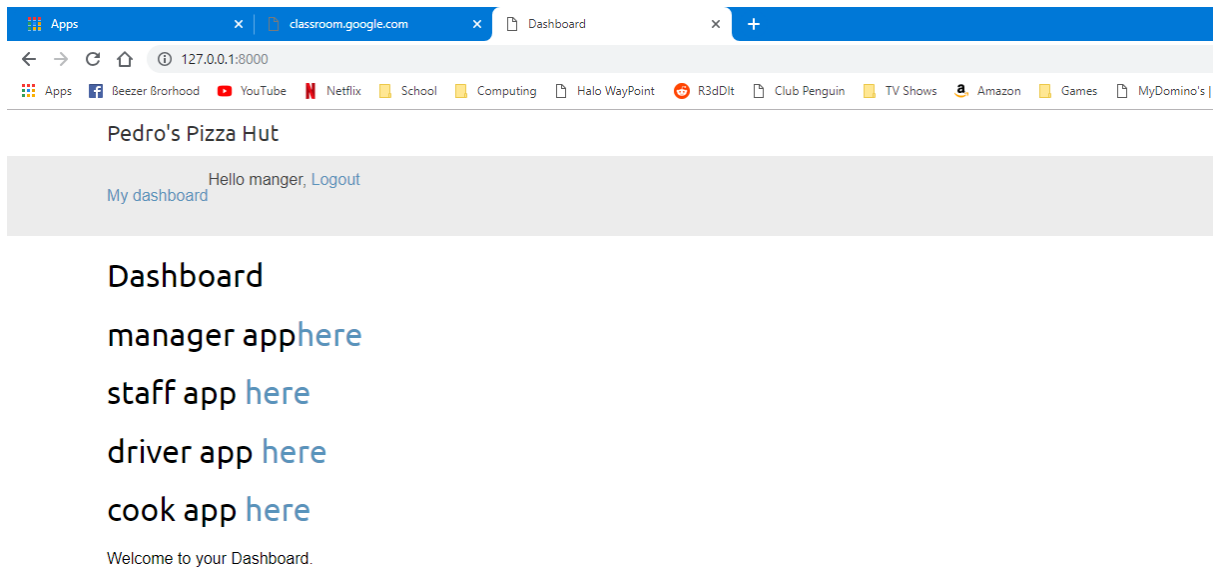


This shows that I have fully prototyped the staff being able to view all of the orders and more.

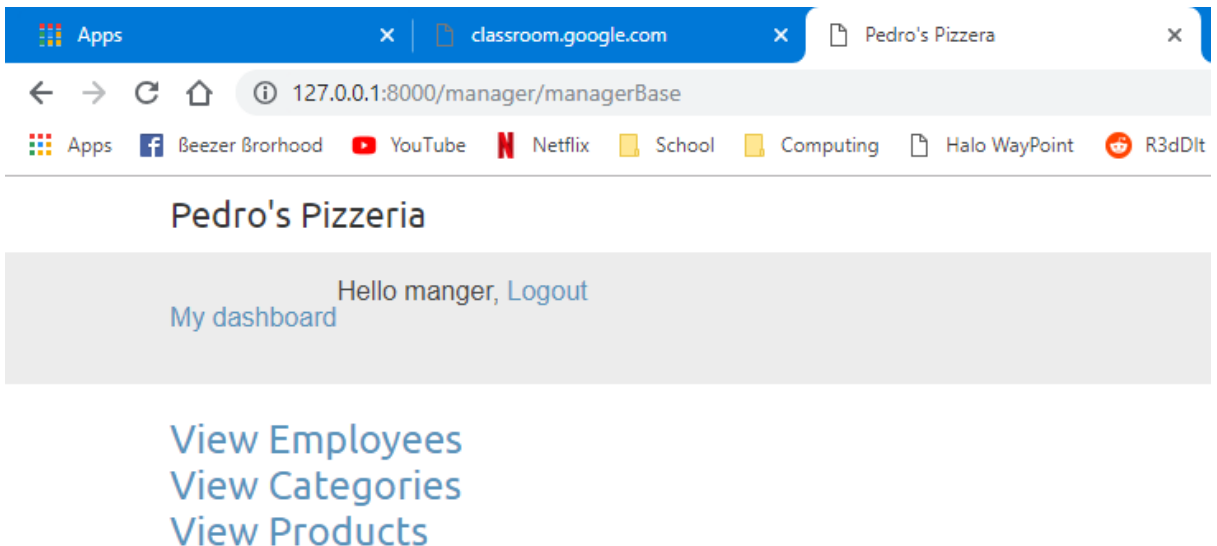
Next, I shall show the manager's view prototyped.

The manager has access to all apps but I will only show the manager app as the rest have already been shown.

When an account with manager access level logs in, their dashboard is shown on the next page.



Here is the manager's app. The manager can view, delete and make some edits to the employees,



categories and products. The manager can also create new categories and products from their respective pages. The manager can also reorder the categories and products like other tables shown before.

Below is the view employees page. From here the manager can view all the details on the employees. The reason the first name and last name attributes do not have real first & last names is because all of the accounts on the system are simply test accounts.

The manager can delete an employee's account by clicking on the delete link. That will delete the employee from the database and then reload the page.

The manager can change the access level of all of the employee's between staff, cook or driver. This will then give the different accounts different access to the different Django apps.

If the manager clicks on the "Update" button a Django form will open and the manager can select from multiple attributes in a drop down table to reorder the employees by.

Pedro's Pizzeria

Hello manger, [Logout](#)

[My dashboard](#) [Manager Menu](#)

## Employees

[update](#)

User ID	First Name	Last Name	Last Login	Address	Staff	Cook	Driver	Delete
4	staff		March 28, 2019, 9:43 p.m.		True <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	<a href="#">Delete</a>
6	cook		Feb. 12, 2019, 11:47 p.m.		False <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	<a href="#">Delete</a>
7	driver		March 28, 2019, 9:25 p.m.		False <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	<a href="#">Delete</a>
8	manger		March 29, 2019, 8:46 a.m.		True <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	<a href="#">Delete</a>

Here is a screenshot showing the ordering of the employees working. The manager can order by date joined, last login or by id.

Pedro's Pizzeria

Hello manger, [Logout](#)

[My dashboard](#) [Manager Menu](#)

## Employees

AttributeChosen: [last\\_login](#)

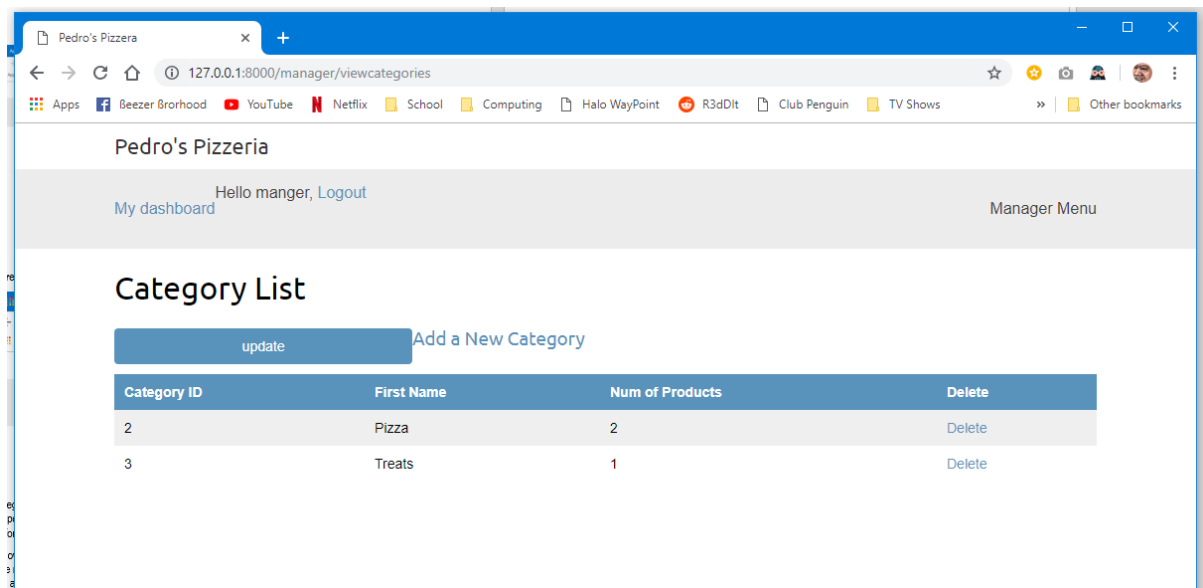
OrderOfCustomers: [Descending](#)

[update](#)

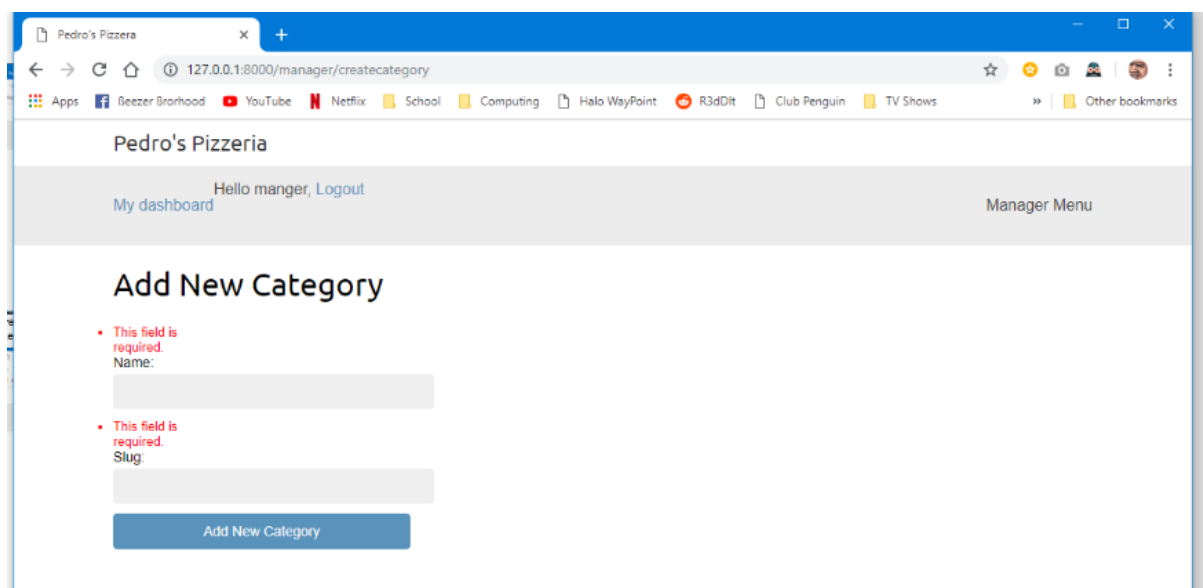
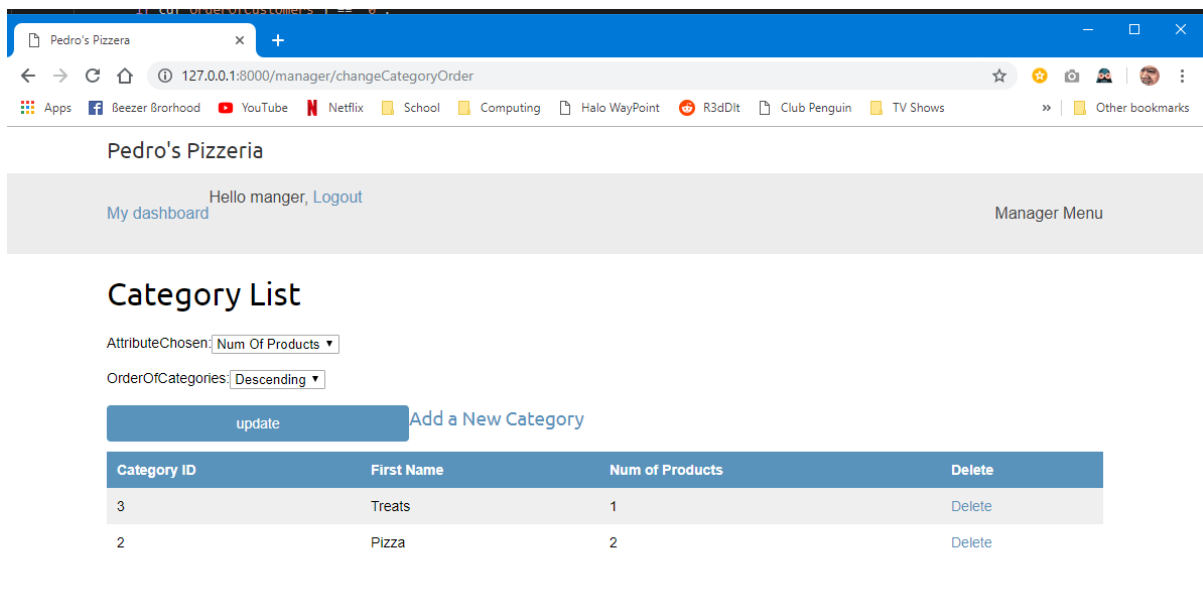
User ID	First Name	Last Name	Last Login	Address	Staff	Cook	Driver	Delete
8	manger		March 29, 2019, 8:46 a.m.		True <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	<a href="#">Delete</a>
4	staff		March 28, 2019, 9:43 p.m.		True <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	<a href="#">Delete</a>
7	driver		March 28, 2019, 9:25 p.m.		False <a href="#">Change Access Level</a>	False <a href="#">Change Access Level</a>	True <a href="#">Change Access Level</a>	<a href="#">Delete</a>

Below is the page for viewing categories for the manager. From here the manager can view basic details on each category in a table, such as name and number of products. The manager can also delete a category from here by clicking the “delete” link. That will delete the category from the database and then reload the page.

The manager can click on the “Add a New Category” link to be brought to a new page where they can add a new category or they can click on the “Update” button to reorder the categories



Here is the reordering of the categories by number of products in descending order.



Above you can see the page that the manager would use to add a new category to the system.

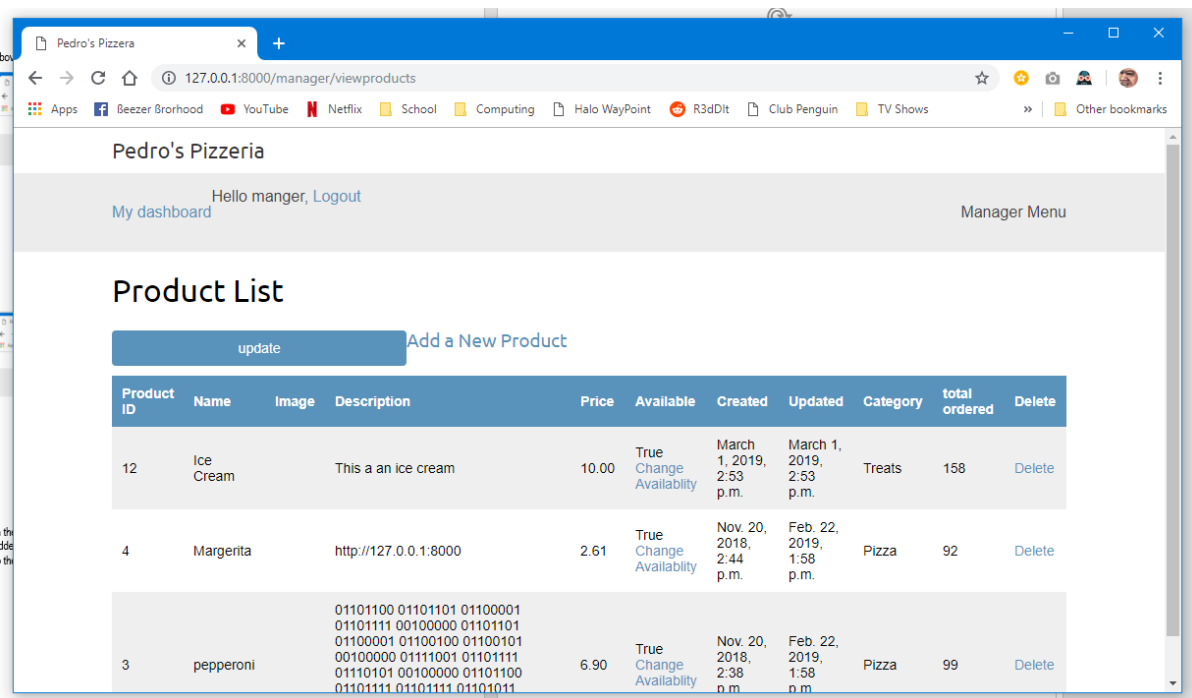
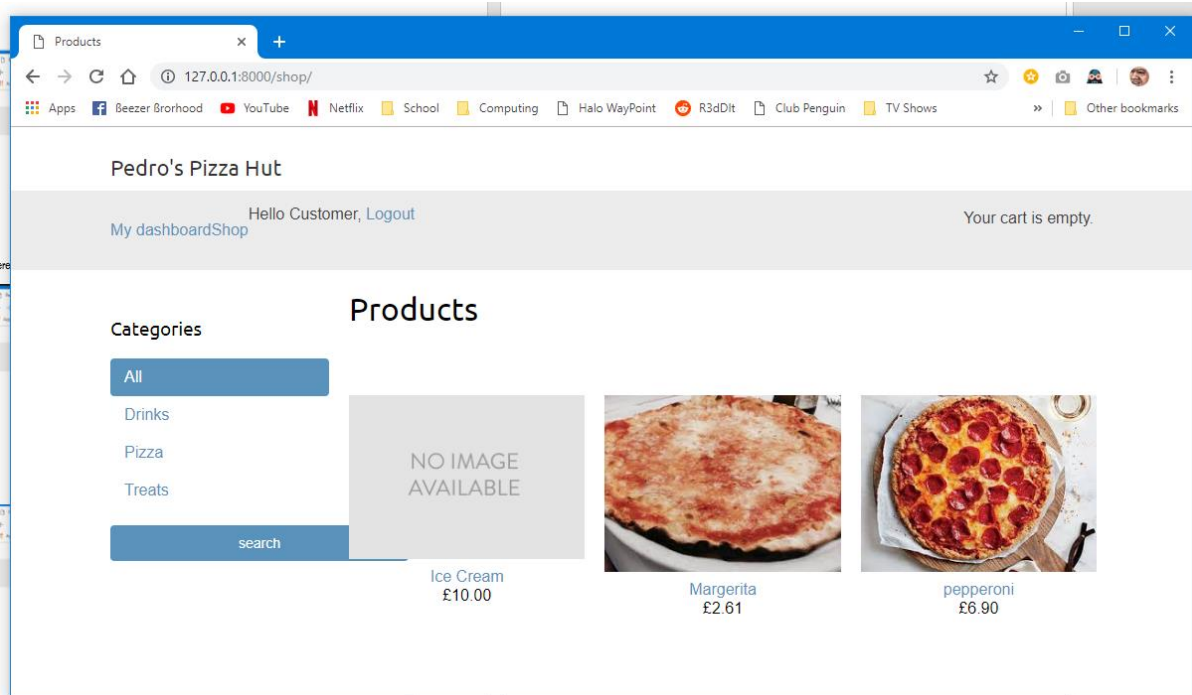
The screenshot shows a web browser window with the URL `127.0.0.1:8000/manager/createcategory`. The page header for 'Pedro's Pizzeria' includes a 'My dashboard' link, a greeting 'Hello manger, Logout', and a 'Manager Menu' link. The main heading is 'Add New Category'. There are two input fields: 'Name:' with the value 'Drinks' and 'Slug:' with the value 'drinks'. Both fields have a red error message: 'This field is required.' Below the fields is a blue button labeled 'Add New Category'.

The screenshot shows the 'Category List' page. It features a table with three columns: 'Category ID', 'First Name', and 'Num of Products'. There are also buttons for 'update' and 'Add a New Category'. The table contains three rows of data: 'Drinks' (ID 16, 0 products), 'Pizza' (ID 2, 2 products), and 'Treats' (ID 3, 1 product). Each row has a 'Delete' link.

Category ID	First Name	Num of Products	Delete
16	Drinks	0	Delete
2	Pizza	2	Delete
3	Treats	1	Delete

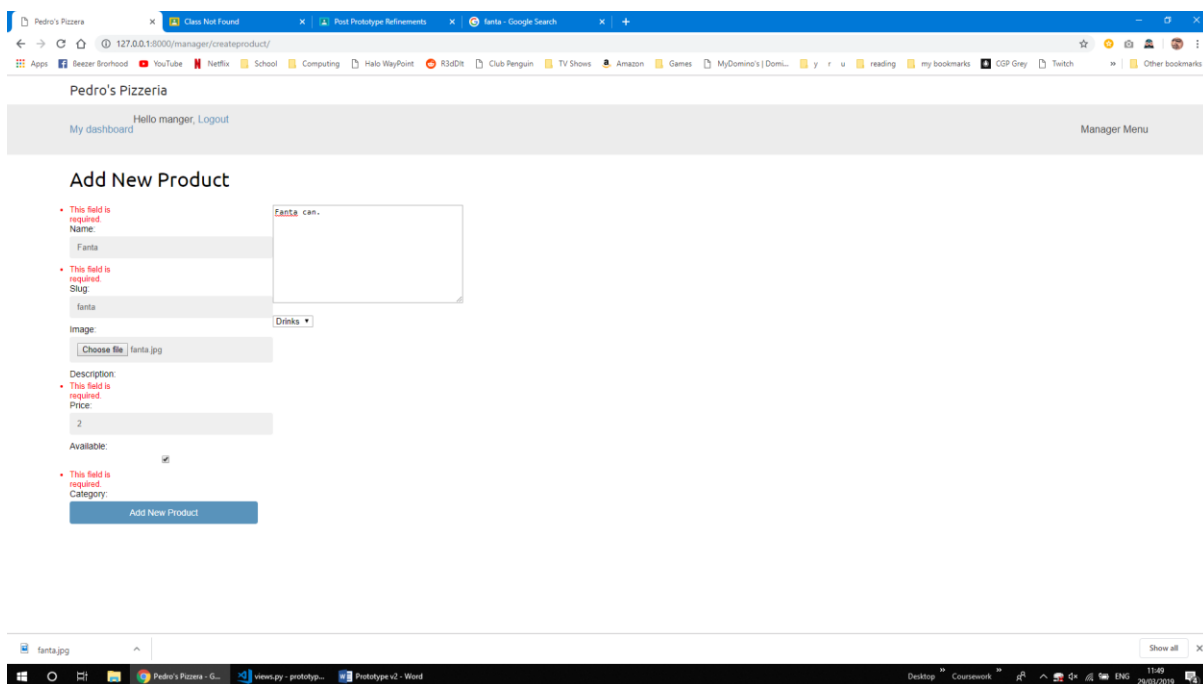
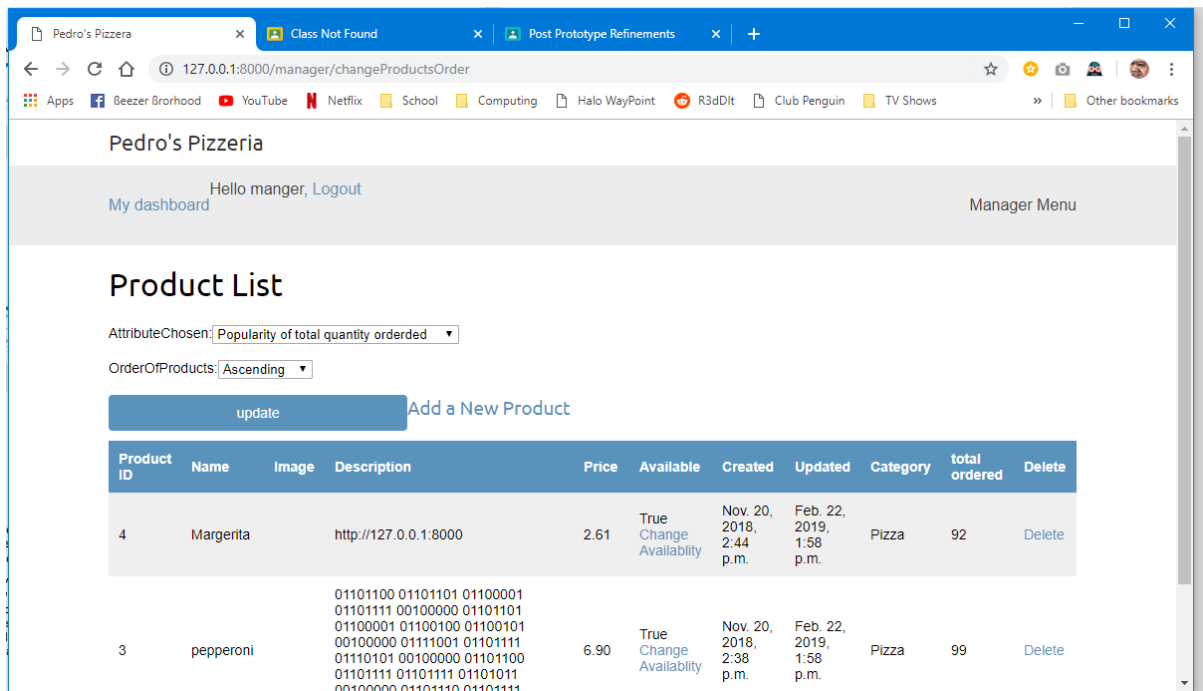
In the above two screenshots and the one below, you can see that the manager has successfully added a new category that has now appeared in the view categories page and when the customer goes to the shop it will appear there too.

Below that is the view products page for the manager's view. Here the manager can click the "Add a New Product" link to go to a different page to add a new product or they can click the "Update" button to reorder the products table by an attribute chosen from a drop down menu. The manager can hide the items from the customer's view by changing the availability of the item by clicking on the "Change availability" link. The manager can also delete the product by clicking the "delete" link. Once the manager has made a change the page is reloaded.



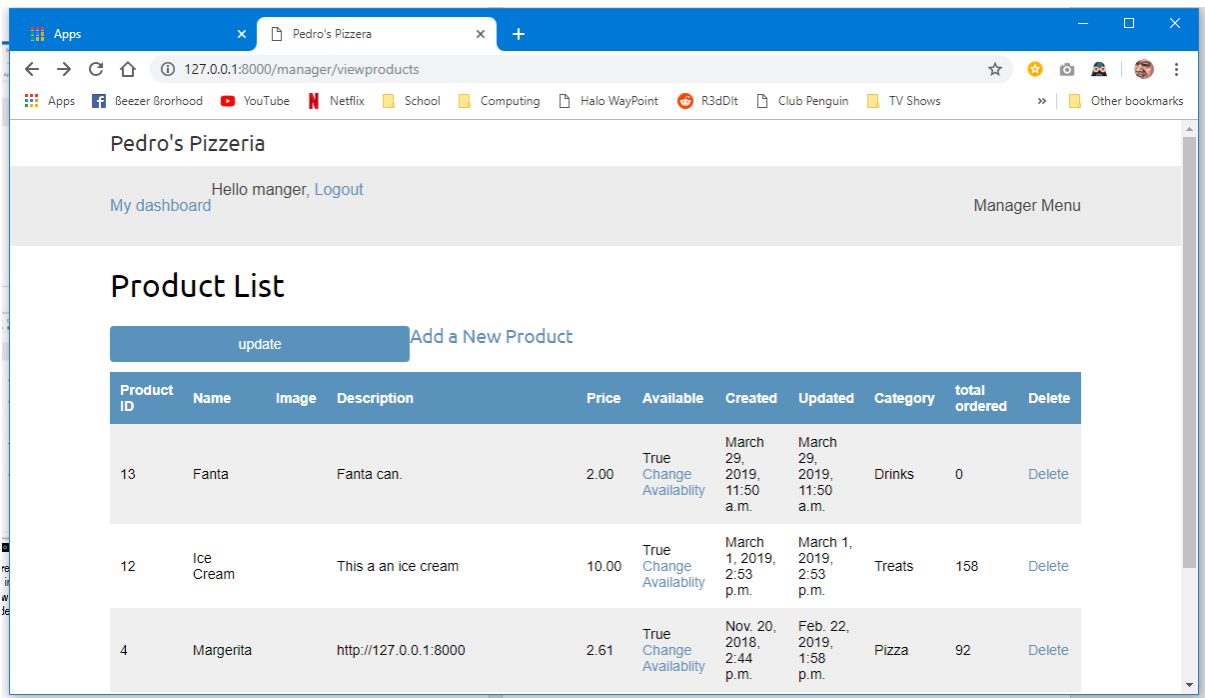
Below you can see the product list being reordered.



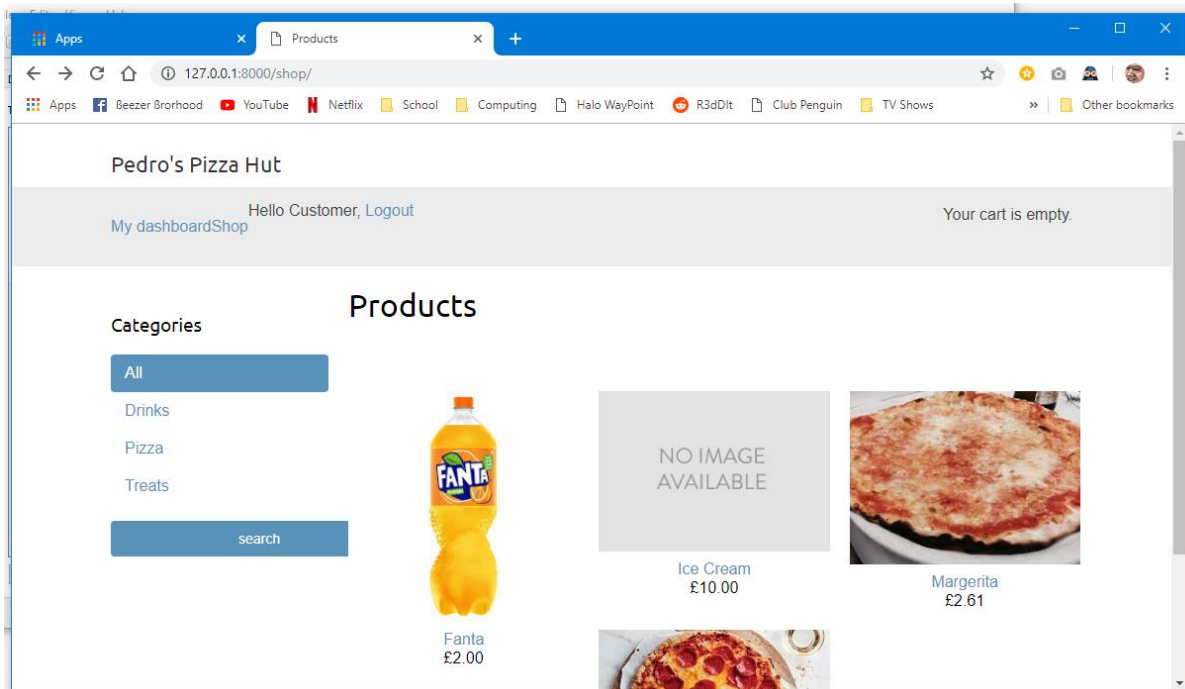


Here is the add a new product page. Here the manager will input the details of the product, including the image file and then press "Add a new Product". Once this happens the manager will be taken to the view products page and there they will be able to see that the new product has been successfully added.

Below is the screenshot after adding the item in the previous screenshots.



Below is a screenshot from the customer's view, showing that the product has been added successfully.



## Evaluation

I evaluated my prototype by comparing it to my objectives & success criteria from my investigation chapter.

The first strength is that the customer can browse the menu in my prototype, and add these items to a cart, switch between multiple categories of the menu and click on each item and view more details about that item.

The second strength is that the customer can view & interact with the cart, and view details of what is in their cart, change the quantity of each item, remove items, see a subtotal & total. However, a weakness of my prototype is that the customer cannot save their order.

The next strength that the prototype has a section for the customer to check out. The customer can access their cart & view their order and the check out page can access to the customer's cart, save information in the customer's cart in the database in the ORDER & ORDERPART tables. The customer can also input their choice of order type (collection, in store or delivery). A weakness of my prototype is that the customer cannot pay with card through this page and they cannot choose to pay with card or cash.

A weakness of my prototype is that the customer cannot track their order's status. Another weakness is that the customer cannot view their previous orders or saved orders.

A weakness of my prototype is that the staff, cooks or drivers are not able to receive SMS messages about their next shifts. Also, there is no prototype for the cooks or drivers

A strength of my prototype is that the staff can view, search, order and delete the customer account, however, a weakness is that they cannot edit them in the prototype.

Another strength of my prototype is that the staff can view the status of each on going order, and each order part. However, a weakness is that they cannot see the target times. Also they cannot associate an order they made for a customer.

Some more strengths of the prototype are that the manager can disable/ enable employee accounts that have already been created, They can edit the menu by editing the Products & Categories tables, and they can hide items. A weakness of the prototype is that the manager cannot edit any information on the restaurant, for example opening times

Another weakness is that the manager cannot set up delivery zones or target times for items. Also, the manager cannot view any reports in the prototype.

Some specific things I can do for improvement would be to add target times. This would only require an extra attribute in the product database and for the manager to input it when creating a new product. ]

Another strength is that as soon as the customer creates an order, they are emailed a notification of it being a success with a receipt. They will also be bale to view a pdf receipt. Another strength is that the staff can view pdfs of different combinations of customer data, their orders data and their order items data. Another strength is that the driver can view the customer's addresses, a pdf of their order's receipt and for each order item they can view a pdf ticket with all of the relative information on it. The manager can also view all of these pdfs.

Another good feature is that I have shown some of the data processing capabilities for the staff and manger's views, such as the average order per week of a customer.

Another improvement that I could make is that is based on features implemented, is that I could email the customer their receipt as a pdf as well as sending them a text based receipt through email.