

# Canteen App 12 DTP

**Final Goal: To create a human-friendly app for school staff and students so they can check out items in the canteen more easily with photos.**

Colour selection:

#004d41

#00000000

#000000

#1ba084

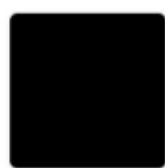
PREVIEW



PREVIEW



PREVIEW



PREVIEW



Font selection:

Trebuchet MS

abcdefghijklmnopqrstuvwxyz

ABCDEFGHIJKLMNPQRSTUVWXYZ

Arial aaaa

ABCDEFGHIJKLMNPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

ABCDEFGHIJKLMNPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxyz

0123456789 01234567890

Draft:

**Burnside High School Canteen**

Welcome to the Burnside High School Canteen! Here you can check items in our canteen and give suggestions and ratings back!  
Enjoy!

[Log In](#)

[Hot food](#)

[Drinks](#)

[Cold Food](#)

[More...](#)

Burnside High School  
Te Kura o Waimairi-iri

30 Greens Road  
PO Box 29677  
Christchurch 8440, New Zealand

To contact email [ [my email...](#) ]

**Burnside High School Canteen**

**Hot food**

Butter Chicken Pie

Garlic Bread

Hashbrown

Burnside High School  
Te Kura o Waimairi-iri

30 Greens Road  
PO Box 29677  
Christchurch 8440, New Zealand

To contact\_email [ [my email...](#) ]

**Burnside High School Canteen**

**Butter Chicken Pie**

A fusion dish that combines the flavors of Indian butter chicken with the classic pastry crust of a pie. It typically consists of tender chicken cooked in a creamy, mildly spiced tomato-based butter chicken sauce, encased in a flaky, golden pastry.

Price: \$7.50

Gluten Free: No  
Vegan: No  
Vegetarian: No  
Sugar Free: No

Burnside High School  
Te Kura o Waimairi-iri

30 Greens Road  
PO Box 29677  
Christchurch 8440, New Zealand

To contact\_email [ [my email...](#) ]

Goal: To learn how to display items works and how to put them into the doc

First step: Ask permission to canteen staff if I can make a website/app about the canteen.

Comment From Canteen Staff: "Yeah I think that's great! We also have notes on foods that are gluten free, sugar free and ect!"

## **Success!**

I also asked Mr Creighton and he liked the idea and gave me permission to make this application.

Things to think about at this point:

- What will I do if there are no photos available? (E.g salads are only available in summer while it is winter right now.)
- What if it takes too long to find the items the user wants? (Scrolling all the way down to the food they want can take a long time.)
- Is the log in system really necessary for now?
- Should I really add a comment system right now?
- Instead of different genres maybe should I make a whole menu where different genres are directed when the button is displayed?

### Making Database:

App I'm using: SQLite Studio

My table name: menu\_items

menu\_items columns:

- id, INTEGER, Primary Key
- name , TEXT, NOT NULL
- price, REAL, NOT NULL
- category, TEXT
- description, TEXT
- gluten\_free, BOOLEAN
- sugar\_free, BOOLEAN
- vegan, BOOLEAN
- vegetarian, BOOLEAN
- image\_link, TEXT

All the BOOLEAN parts are set to 0.

categories columns:

Nothing for now... Because I'm trying to make the simplest database for now, I might change the categories to many to many relationships with the menu\_items columns.

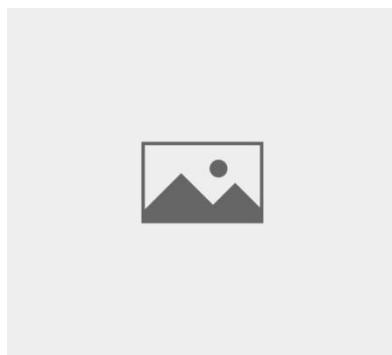
Inserting information into data in each column:

Extremely boring process, I had to use this sql code to insert everything in:

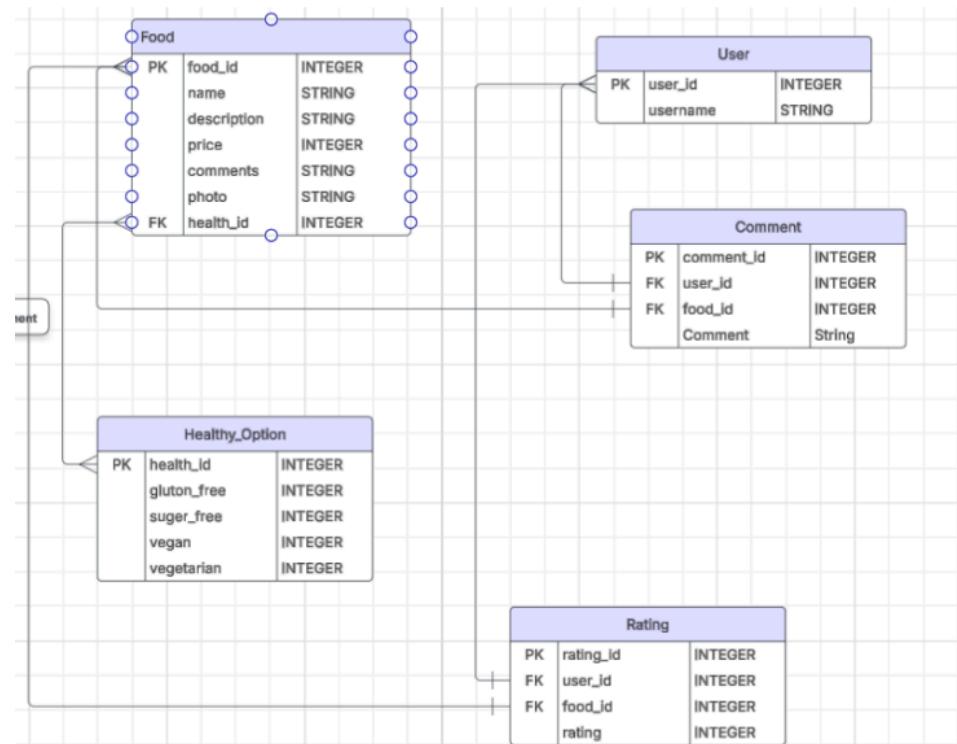
```
INSERT INTO menu_items (id, name, price, category, image_link)
VALUES (1, 'Hashbrown', 1.00, 'Hot Food (every day)', '/static/Images/Canteen_App_Food/Hash_Brown.JPG');
```

For the gluten free and other parts I turned them to 1 if the item is actually gluten free/sugar\_free ect.

If there are no photo in the food, I replaced with this photo in my static/image folder:

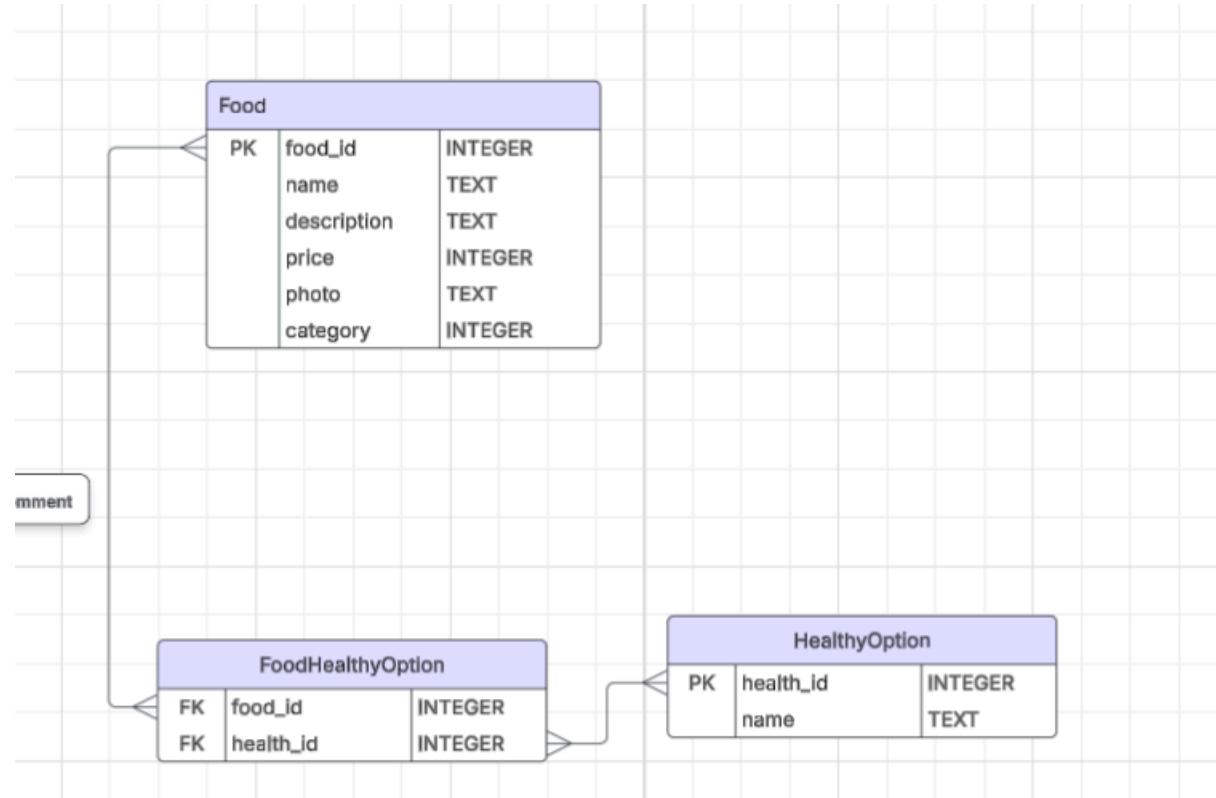


ER Diagram at the beginning of planning:



This is getting changed because I realised I do not need a log in system in a menu list as the comment feature will not be in my website.

I also decided to remove the many to many relationship I mentioned before from the category because 1 category can go to many foods but one food can only have one category so a separate table is not needed. But I could get a many to many relationship with healthy options since one healthy option can go to many foods and one food can have many healthy options.



This is my new ER diagram because one food might have many healthy options such as vegan foods automatically identified as vegetarian

If I did not insert any photos then the code and layout of my website would be unorganized and users would find it difficult to look at the data.

After checking the chart I realized I did something wrong like the name should be text along with other columns not string.

Here is my initial completed data:

<b>id</b>	<b>name</b>	<b>price</b>	<b>category</b>	<b>description</b>	<b>gluten_fr</b>	<b>sugar_fre</b>	<b>vegan</b>	<b>vegetaria</b>	<b>image_link</b>
1	1 Hash Brown	1	Hot Food (every day)	Hot, crispy hash browns made from potatos.	0	0	0	0	/static/Images/Canteen_App_Food/H
2	2 Miso Soup	2.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/M
3	3 Garlic Bread	3	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/G
4	4 Nachos	7	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/N
5	5 Noodles (Chicken)	3.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/N
6	6 Noodles (Beef)	3.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/B
7	7 Noodles (Hot & Spicy)	4.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/H
8	8 Noodles (Shin Cup)	4	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/N
9	9 Pies (Mince)	5.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/P
10	10 Pies (Mince/Cheese)	6	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/P
11	11 Pies (Steak/Cheese)	6	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/P
12	12 Pies (Butter Chicken)	6	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/P
13	13 Bacon & Egg Pie (Slice)	4	Hot Food (every day)	NULL	0	0	0	0	/static/Images/No_Photo.jpg
14	14 Vegan Pie (Kumera & Cashew)	6.5	Hot Food (every day)	NULL	0	0	1	1	/static/Images/Canteen_App_Food/P
15	15 Vegan Pie (Mexican)	6	Hot Food (every day)	NULL	0	0	1	1	/static/Images/Canteen_App_Food/P
16	16 Spinach & Ricotta Roll	5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/No_Photo.jpg
17	17 Sausage Roll	3	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/S
18	18 Panini	5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/P
19	19 Cheese, Ham & Pineapple Toastee	4	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/T
20	20 Iced Coffee	3.5	Cold Drinks	NULL	0	0	0	0	/static/Images/Canteen_App_Food/I
21	21 Steamed Buns	5.5	Hot Food (every day)	NULL	0	0	0	0	/static/Images/Canteen_App_Food/S
22	22 Rolls	7	Sandwiches/Rolls/Wraps	NULL	0	0	0	0	/static/Images/No_Photo.jpg
23	23 Wraps	7	Sandwiches/Rolls/Wraps	NULL	0	0	0	0	/static/Images/No_Photo.jpg

This is only some sections but basically done. I will add the descriptions later. But the basic database is done.

One thing I don't get is why programmers put images and css files into a static folder. So I need to do some research to find the reason.

### Conclusion:

Programmers uses static folder for many reasons:

- Faster processing speed for web servers or develop servers built into frameworks such as flask which is what I am using.
- By isolating the static files it can prevent random access from other users to sensitive the application data or code which in this case my images and style.css.
- Flask needs to know that css and images are not dynamic but static so by labeling a static folder Flask is able to recognise which files are static and which are dynamic.

Database comments:

Name	Comment
Zach Newman (Friend)	"You should add data to the health table as a many to many relationship."
Mum	"Nice menu but you should add descriptions so users are able to read the descriptions."

Overall review/working on in the future:

- Add descriptions to the foods

- Change the healthy option column as a separate table and as a foreign key with a many to many relationship.

After some time this is my final database:

## Menu\_items table:

FoodHealthyOption table:

## HealthyOption table:

health_id	name
1	Gluten Free
2	Sugar Free
3	Vegetarian
4	Vegan

Testing ( To see if the database meets the users' requirements.)

What	How	Expected	Outcome	Notes
Show all the information on the menu_item table.	SELECT * FROM menu_item	All the items shown with the information	All data is displayed apart from descriptions which are not done.	This is just basic code to show all information which tests that my database works
To display all the data users wanted to see	By using the first query bellow (Too big can not fit in the column)	To show name, description, price, category and all healthy options in items.	All the data mentioned in the expected column showed up.	Meaning that this works and might be used on my specific item details page in the menu website.
To show all items that is vegan	By using the second query bellow	To show all the items that are vegan	All items that are vegan are shown.	This works very well.

```

SELECT
    f.name,
    f.description,
    f.price,
    f.category,
    GROUP_CONCAT(ho.name, ', ') AS healthy_options
FROM menu_items f
LEFT JOIN FoodHealthyOption fho ON f.id = fho.food_id
LEFT JOIN HealthyOption ho ON fho.health_id = ho.health_id
GROUP BY f.id;

```

```

SELECT f.name
FROM menu_items f
JOIN FoodHealthyOption fho ON f.id = fho.food_id
JOIN HealthyOption ho ON fho.health_id = ho.health_id
WHERE ho.name = 'Vegan';

```

Now I need to make sure the information matches the actual application and real world:

My Sql database:

<u>id</u>	<u>name</u>	<u>price</u>	<u>category</u>	<u>description</u>	<u>image_link</u>
1	Hash Brown		1 Hot Food (every day)	Hot, crispy hash browns made from potatos.	/static/Images/Canteen_App_Food/Hash_Brown.JPG
2	Miso Soup	2.5	Hot Food (every day)	A warm, savoury Japanese soybean broth with a light, comforting flavour.	/static/Images/Canteen_App_Food/Miso_Soup.JPG
3	Garlic Bread		3 Hot Food (every day)	Toasted bread topped with garlic butter, extremly soft inside.	/static/Images/Canteen_App_Food/Garlic_Bread.JPG
4	Nachos		7 Hot Food (every day)	Corn chips topped with melted cheese, flavoured beans, and sour cream.	/static/Images/Canteen_App_Food/Nacho.JPG

<u>food_id</u>	<u>health_id</u>
1	3
4	3

<u>health_id</u>	<u>name</u>
1	Gluten Free
2	Sugar Free
3	Vegetarian
4	Vegan

When I use this sql query on my item detail webpage all the information on the database should show.

```

SELECT
    f.name,
    f.description,
    f.price,
    f.category,
    GROUP_CONCAT(ho.name, ', ') AS healthy_options
FROM menu_items f
LEFT JOIN FoodHealthyOption fho ON f.id = fho.food_id
LEFT JOIN HealthyOption ho ON fho.health_id = ho.health_id
GROUP BY f.id;

```

Website:

### Hash Brown



Price: \$1.0

Description: Hot, crispy hash browns made from potatos.

Healthy Options: Vegetarian

### Nachos



Price: \$7.0

Description: Corn chips topped with melted cheese, flavoured beans, and sour cream.

Healthy Options: Vegetarian

---

### Miso Soup



Price: \$2.5

Description: A warm, savoury Japanese soybean broth with a light, comforting flavour.

No healthy options available.

### Garlic Bread



Price: \$3.0

Description: Toasted bread topped with garlic butter, extremly soft inside.

No healthy options available.

[Back to Menu](#)

And the id matches as well:

127.0.0.1:5000/item/1

chool Bio Predator Free R...

de school o ri-ri

## Burnside High School Canteen

### Hash Brown



127.0.0.1:5000/item/2  
school Bio Predator Free R...  
de  
chool  
a o  
iri-ri

## Burnside High School Canteen

Miso Soup



Price: \$2.5

And because it Burnside High School canteen menu, so it definitely matches the actual canteen menu:

BURNSIDE HIGH SCHOOL – CANTEEN MENU		
Term 1, 2025		
HOT FOOD (every day)		SALADS (Summer only)
Hash browns	1.00	Salad (variety daily)
Misco soup	2.50	
Garlic Bread (large)	3.00	
Nachos (beans, cheese & sour cream)	7.00	HOT FOOD (Different meal each)
Noodles (chicken, beef)	3.50	Macaroni cheese
Noodles (hot & spicy)	1.50	Baked potato (beans/cheese/sour cre

Relevant Implications:

My data stores information about the canteen menu. This does not get any users' information which gives them privacy and the data in my implication is accurate because I asked the canteen staff about the healthy options and also did research on products that the canteen staff doesn't know about. I made sure to ask the canteen staff and Mr Creighton about this idea before making the application. I am not just getting information out of nowhere. I have also been chatting with the canteen staff about the application to gain advice and respect their choices.

## Coding:

App I am using: Visual Studio Code ( Extremely easy to use and easy to commit to github as well.)

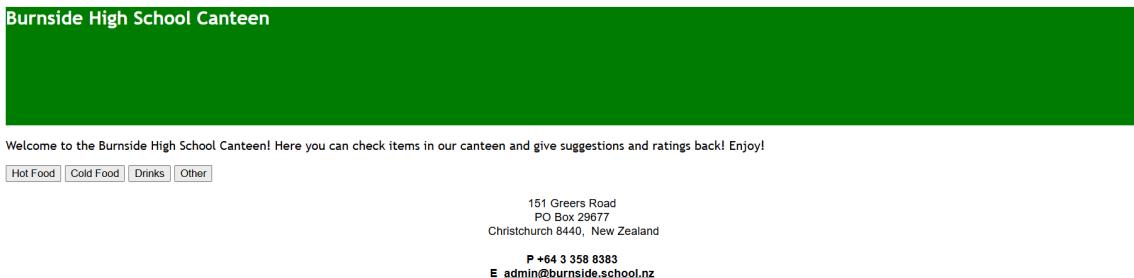
Things I need to include in my code:

- Html

- Python
- Flask
- Jinja2
- Css
- Javascript? (Maybe I need it? Not sure right now)

First I decided to get html files ready for my different genres:

My home page so far:



Definitely not the final layout, but it's a good starting point.

Things I thought about after this:

- More buttons will be needed and customised to reach the user's max satisfaction when using this app/website.
- The footer can be at the very bottom and instead of BHS information it could just have my email address and name so if there are problems it can direct to me.
- The overall layout can be improved

This is my second try at making the home page:

# Burnside High School Canteen



Welcome to the Burnside High School Canteen Menu!

Hot Food (every day)

Cold Drinks

Sandwiches/Rolls/Wraps

Muffins/Slices/Cookies/Cakes

Potato Chips

Salad (Summer only)

Hot Food (Different Meal Each Day)

Ice Creams

Fruit & Yoghurt

Hot Drinks

In my opinion this is much better than the previous one as it is more clean and obvious about what is on the menu.

But I still need to add a footer and directions to the buttons.

Making the menu website (Where the button leads):

```
#Route to show all hot foods available in food database.
@app.route('/hot_food')
def all_hot_food():
    conn = sqlite3.connect('food.db')
    cur = conn.cursor()
    cur.execute('SELECT * FROM menu_items WHERE category="Hot Food (every day)"')
    hot_food = cur.fetchall()
    conn.close()
    return render_template('menu.html', hot_food = hot_food)
```

This is able to show all the information from the hot food category.

Then I realised this is not very productive as I need to do the repeated codes for all the categories in my menu ( 10 in total for now!) so instead of this route I changed it to something like this:

New improved route:

```

def get_menu_data():
    conn = sqlite3.connect("food.db")
    # row factory to access columns by name
    conn.row_factory = sqlite3.Row
    cur = conn.cursor()

    # get all the food categories in the database
    cur.execute("SELECT DISTINCT category FROM menu_items")
    categories = cur.fetchall()

    # get the menu items from categories
    menu = []
    for cat in categories:
        category_name = cat["category"]
        cur.execute(
            "SELECT id, name, price, image_link FROM menu_items "
            "WHERE category = ?",
            (category_name,))
    items = cur.fetchall()

    # turn each item from Row to dictionary
    menu_list = []
    # creates a loop to repeat each item
    for item in items:
        menu_list.append({
            "id": item["id"],
            "name": item["name"],
            "price": item["price"],
            "image_link": item["image_link"]
        })
    # Changes category name to a valid name for HTML id
    menu.append({
        "id": (
            category_name.lower()
            .replace(" ", "_")
            .replace("(", "")
            .replace(")", ""))
        ,
        # Display name
        "name": category_name,
        # Display menu items
        "menu_items": menu_list
    })
conn.close()
return menu

```

Here instead of just hot food for example I repeated every category with a loop and also turned all the data from row to dictionary. From my friend Kevin and Google I learnt that by turning from row to dictionary it provides a more clear way of showing structured data. It is also extremely easy to access the data by name so in this case the id, name, price, and image\_link is extremely useful to use in the future codes.

For example:

Instead of doing something like this:

```
{% for f in hot_food %}  
    <div class="card">  
          
        {{f[1]}}  
        <br>Price: ${{f[2]}}
```

I could do something like this:

```
{% for item in category.menu_items %}  
    <div class="item">  
          
        <h3>{{ item.name }}</h3>  
        <p class="price"> Price: ${{ "%2f" |format(item.price) }}</p>  
    </div>  
</a>  
{% endfor %}
```

This is way more clean , understandable and easy to use the variables again when I need to. This is also simplified iteration and processing by allowing easy access to field values within the loop which simplifies processing logic.

When I saw the KFC menu, I loved how they changed the url when users scroll down to see the whole menu. I also like how they can click on top of the section bar to go to the genre like this:

kfc.co.nz/menu/individual-meals

BHS School Bio Predator Free R...



**MENU** CATERING FIND A KFC DEALS KFC FOR GOOD

**Start Order** Sign In

< TO SHARE **INDIVIDUAL MEALS** BURGERS LIGHTER EATS SNACKS SIDES DRINKS & DESSERTS CATERING

### INDIVIDUAL MEALS



**Secret Recipe Katsu Bowl**

A sliced Secret Recipe Fillet, Jasmine Rice, crunchy Slaw, Japanese Mayo and Teriyaki Sauce

**Add**



**Zinger Katsu Bowl**

A sliced Zinger Fillet, Jasmine Rice, crunchy Slaw, Japanese Mayo and Chilli Relish

**Add**



**Wicked Box**

4 Wicked Wings, reg Chips, reg Potato & Gravy and a Bread Roll

**Add**



**Hot & Crispy Boneless Meal**

3pcs Hot & Crispy Boneless, reg Chips, a Bread Roll and reg Potato & Gravy

**Add**

After some research I found out that this feature can be done by using javascript. Javascript code is able to enhance user interfaces such as changing url when scrolling through a category by creating an observer and changing the url when the observer detects a new category is scrolled.

So I decided to change my plan and do something similar like this:

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

Search items here...



**Bacon & Egg Pie (Slice)**

\$4.00



**Cheese, Ham & Pineapple Toastee**

\$4.00



**Garlic Bread**

\$3.00







Here I used javascript for both search bar and changing url when scrolling through the page which I just learnt a new skill for coding ( Still not that good at it).

I learnt it first by watching youtube videos about javascript and looking at the W3schools website for some help with code. I also found [this website](#) useful for changing the url. And with help from a relative who is pretty confident at javascripts and websites I am finally able to get both features for the application.

Here is the javascript code I concluded with:

For search bar:

```
<script>
    // Search bar filter
    document.getElementById('searchBar').addEventListener('input', function(e) { // Get input from search bar
        const search = e.target.value.toLowerCase(); // Converts the words to lowercase
        document.querySelectorAll('section').forEach(section => {
            let visible = false;
            section.querySelectorAll('.item').forEach(item => { // grabs all items
                const name = item.querySelector('h3').textContent.toLowerCase(); // gets the name of the menu item and turns it lowercase

                // Check if the name is the same as the search text
                if (name.includes(search)) {
                    item.style.display= ''; // shows the item if it matches
                    visible = true;
                }
                else {
                    item.style.display = 'none'; // hides the item if it doesn't match
                }
            });
            section.style.display = visible ? '' : 'none';
        });
    });
});
```

Here you can search items for specific items in the menu and even if users search up not in correct capital letter form results still show up. There are still some testing I need to do so I set up a chart:

For changing url:

```

//grabs all sections and nav links
const sections = document.querySelectorAll('section');
const navLinks = document.querySelectorAll('nav a');

//creates an intersection observer to watch the sections
const observer = new IntersectionObserver((entries) => {
    entries.forEach(entry => {
        if (entry.isIntersecting) {
            const id = entry.target.getAttribute('id');
            // Change website title to current genre/category name
            const heading = entry.target.querySelector('h1');
            if (heading) {
                document.title = heading.textContent + ' | Canteen Menu';
            }
            // Change the URL to reflect the current genre/category
            const newurl = `/menu/${id}`;
            if (window.location.pathname !== newurl) {
                window.history.replaceState(null, '', newurl);
            }
        }
    });
}, { threshold: 0.15 });

sections.forEach(section => {
    observer.observe(section);
});

```

But there is currently a bug where when I reload the page the url shows that it is not found. I think it is because I didn't give it a function in the routes.py so that is why the url is not changing.

After debugging I realised the problem:

Because I am changing the url in /menu page, when I try to refresh /menu/hot\_food for example I am not actually refreshing the /menu page but a new url which is /menu/hot\_food. So I need a way to fix this.

I decided to change the url from /menu/\${id} to /menu#\${id} because hash urls are handled by the browser so reloading will not cause a 404.

Another way I could have done this is create a route for every genre in my category but I feel like it would have repeated codes which is not what I want.

Here is my new coding:

```

//grabs all sections and nav links
const sections = document.querySelectorAll('section');
const navLinks = document.querySelectorAll('nav a');

//creates an observer to watch the genres/categories
const observer = new IntersectionObserver((entries) => { // callback when intersection changes
    // loop
    entries.forEach(entry => {
        // check if the genre is intersecting
        if (entry.isIntersecting) {
            const id = entry.target.getAttribute('id'); // get the id of category
            const heading = entry.target.querySelector('h1');// get the heading of category
            // change the website title to reflect the current category
            if (heading){
                document.title = heading.textContent + ' | Canteen Menu';
            }
            // change the URL to reflect the current genre/category
            const newurl = `/menu#${id}`;
            if (window.location.hash !== `#${id}`) {
                window.history.replaceState(null, '', newurl);
            }
        }
    });
}, { threshold: 0.15 }); //if 15% of the genre is scrolled then the code above runs

// checking each category to see if it is intersecting
sections.forEach(section => {
    observer.observe(section); //observe each category
});

```

Now whenever reloading on the menu page the code doesn't break or doesn't find the url which fixed the bug.

Testing Process:

What	How	Expected	Outcome	Notes
To go to the specific genre on the menu in home page	By clicking on any genre buttons on the home page.	To go to the specific genre in the menu	Sometimes it works, sometimes it doesn't... it just goes to the front of the page without directing to the position of the genre.	I need to work on this as clicking on the genre and not directing on the codes can be annoying and users will find it frustrating.
To direct to a specific genre on the navigation tool on the menu.	By clicking on a section on the navigation tool it should direct to the food genre.	The page goes to the genre section the user clicked.	The page rolls down to the section users clicked which is what users want.	This ended up succeeding which I will keep the feature

Go back to home page	By clicking on the school logo	To go back to home page	Went back to the home page which is what users want.	This ended up successful and I will be keeping this feature.
Prevent overflow error	By inserting values that are too large in the application to break the code.	Direct to 404.html because the value is too large.	Error pops up with overflow error. The code breaks.	This something I need to debug because the code can not break

## Debugging process:

In the first testing process by clicking on the button sometimes it doesn't work as it does not go to the specific genre. But after changing from the / for url to # This bug got fixed as well. I think it is because in my home.html the buttons directs to /menu#[{ category.id }] which doesn't always go to the url users wanted which was /menu/{{category.id}}.

More Testing:

404 error test/boundary:

What	How	Expected	Outcome	Notes
To check if inserting o or id that doesn't exist breaks the code	Typing o and random numbers/letters/special letters in /item/{id}	Directs to 404.html	Went to 404.html	Expected result.
To check if users can break the code from home.html	By typing random variable in home.html works	Directs to 404.html	Goes to another search result/not found	I guess it is alright because it didn't break or anything it just went to another page.
To check users can break code	By typing random things in the	Direct to 404.html?	Nothing happens, the web stays the	I don't think users are actually

from menu page	url.		same	breaking the code so it should be fine
To check if users can abuse the search bar	By typing too much variables in to the search bar	Limited to 30 letters/numbers/special characters	Once it is more than 30 letters the search bar stopped receiving variables	Expected results

home.html testing:

What	How	Expected	Outcome	Notes
To see if the colour changes when button is pointed	By moving cursor to one of the buttons	The colour changes from dark green to light green	The colour changes like the expected.	Expected results and I think this addition is pretty cool.
To see if the buttons on menu page directs to the specific category in menu_scroll.html	By clicking on every category buttons on the menu page multiple times	Everytime user clicks the button it should lead to the category users clicked.	Tested each category 3 times and everytime it works	Expected results which are pretty good.
To see if clicking on the home page doesn't do anything	By clicking on the logo multiple times	Nothing should happen	Nothing happened	Expected results

menu\_scroll.html testing:

What	How	Expected	Outcome	Notes
To see if navigation button works	By clicking on all of the navigation buttons	Directs to specific category	Tested 3 times each category and works	Expected result and definitely gives the user

			everytime.	a better option instead of scrolling to the specific category.
To see if the buttons directs to details work	By clicking on multiple buttons on the page to see if they all work	All of them directs to their details page	All of them works	This better work otherwise users can not look at the details
To see if search bar works	By searching up items in the search bar	Items that are related to the input should show up	Items showed up without needing capital letter/lower case	This is definitely the hardest function for me to do because of making it human friendly.
If code crashes	when navigation is clicked as an item is searched	Nothing should happen	Nothing did happen only url will change	I don't think any codes were broken so it is fine.

item\_detail.html testing:

What	How	Expected	Outcome	Notes
To see if users can go back home without going back to menu	Clicking on the school logo	Go back to main page	Went back to main page	This is a very important feature as it gives user control and freedom
To see if users can go back to menu without go back button in google	Clicking on button 'back to menu'	Goes back to the menu page	Went to menu page	I think this is also a part of user control and freedom
To check if all	By comparing	Same data	Same data	Important

information is displayed on the page correctly	the data on the page with sql database and actual canteen menu	compared with everything		stuff otherwise I might be giving out false information.
--	--	--------------------------	--	--

## Feedbacks/changes made

After showing my application to many people from my family to teachers I received a lot of feedback for me to consider to make my application better.

From Marshall Hu (Student/friend): "The website is useful but looks too plain and a bit boring. Maybe add more colours and dynamics to make it more interesting."

I thought about this and don't agree with this because apart from students, staff might also use this which might not get used to a range of colours and extra moving effects. Students who are also not used to this might not like this either. So I think staying with this is the best option. Colours are simple but easy to use.

From my grandma (doesn't understand English so have to explain and translate): "That is considerate for your school! I like how it looks like an actual menu and it is big so people can see properly. Maybe make sure when you click on an item you actually know that you are gonna click it. I don't know if you know what I mean."

I think what she meant to say is that when hovering over an item users should actually know that it is being selected if left click is pressed then it goes to that detail. I think this is a great idea so I made some changes here:

```

/* Individual Item style */
.item{
    margin: 12px;
    padding: 12px;
    border: 1px solid #white;
    border-radius: 8px;
    background: #f9f9f9;
    transition: transform 0.5s;
    gap: 24px;
}

.item:hover{
    transform: scale(1.05);
}

```

Here I made that whenever the user hovers the item it gets bigger so users know that it is being selected.

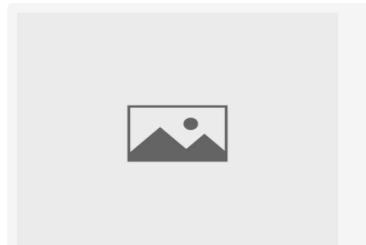
Before being selected:



Cookie Time Cookie  
Price: \$1.80



Gluten-Free Cookie  
Price: \$4.50



Vegan Cookie  
Price: \$4.00

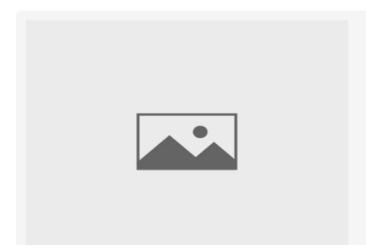
After being selected:



Cookie Time Cookie  
Price: \$1.80



Gluten-Free Cookie  
Price: \$4.50



Vegan Cookie  
Price: \$4.00

It is hard to see but it definitely has gotten bigger which helps users to see it is being selected.

Zac Newman (Programmer): "Everything looks clean! But you should really need a footer. A website needs a footer man otherwise people might steal your website unless you claim no copyright!"

I forgot about that... I'll add it right away:



This is also a good example of help and documentation because people can contact me if they need any help.

Code:

```
<div class="footer">
|   |   <h3>2025 Burnside High School Canteen, all rights reserved
|   |   |   <br> Contact 22011@burnside.school.nz for questions/errors<br></h3>
</div>
```

Kevin (Programmer): "Yeah you should definitely add a comment section and also review stars it will instantly make the application better. A log in system should also be added to make everything look fancy."

This is such a good idea, I had this in mind as well.

But I don't think I have the time right now to go back to sql and add a log in system and comment section. But I will keep this in mind if I have more time. I will definitely consider this as a feature of my application.

## Feature Check

This is my final project:

The screenshot shows a web browser window with the URL 127.0.0.1:5000. The title bar includes the school logo, "BHS", and other tabs like "School" and "Bio Predator Free R...". The main header is "Burnside High School Canteen" with the subtitle "Welcome to the Burnside High School Canteen Menu!". Below the header is a grid of ten buttons arranged in two rows of five. The top row contains: "Hot Food (every day)", "Cold Drinks", "Sandwiches/Rolls/Wraps", "Muffins/Slices/Cookies/Cakes", and "Potato Chips". The bottom row contains: "Salad (Summer only)", "Hot Food (Different Meal Each Day)", "Ice Creams", "Fruit & Yoghurt", and "Hot Drinks".

2025 Burnside High School Canteen, all rights reserved  
Contact 22011@burnside.school.nz for questions/errors

If user hovers on a button it turns colour:

This screenshot is identical to the first one, but the "Cold Drinks" button in the top row is highlighted with a darker teal color, indicating it has been hovered over.

2025 Burnside High School Canteen, all rights reserved  
Contact 22011@burnside.school.nz for questions/errors  
127.0.1.5000/menu#cold\_drinks

If user clicks on it, it goes to the selected category:

← → ⌂ 127.0.0.1:5000/menu#cold\_drinks

BHS School Bio Predator Free R...

# Burnside High School Canteen

Search items here...

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

Iced Coffee  
Price: \$3.50

E2 Large  
Price: \$4.50

Pump Water  
Price: \$3.50

When users click on a button on navigation it changes colour and directs to the specific category:

127.0.0.1:5000/item/51

# Burnside High School Canteen

Search items here...

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

Iced Coffee  
Price: \$3.50

E2 Large  
Price: \$4.50

Pump Water  
Price: \$3.50

← → ⌘ 127.0.0.1:5000/menu#sandwiches/rolls/wraps

BHS School Bio Predator Free R...

**Burnside High School Canteen**

Search items here...

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

**Rolls**  
Price: \$7.00

**Wraps**  
Price: \$7.00

**Vegetarian Wraps**  
Price: \$6.00

And when users search an item in the search bar the results show up without capitals or some errors in typing:

**Burnside High School Canteen**

arlic

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

**Hot Food (every day)**

**Garlic Bread**

**Burnside High School Canteen**

coke

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

**Cold Drinks**

**Coke No Sugar (Small)**  
Price: \$2.00

**Coke No Sugar (Large)**  
Price: \$4.00

By scrolling through the category the url and title should change:

Hot Food (every day) | Canteen

127.0.0.1:5000/menu#hot\_food\_every\_day

BHS School Bio Predator Free R...

**Burnside High School Canteen**

Search items here...

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

## Hot Food (every day)



Hash Brown

Price: \$1.00



Miso Soup

Price: \$2.50



Garlic Bread

Price: \$3.00

This is hot food (everyday) url and title

Cold Drinks | Canteen Menu

127.0.0.1:5000/menu#cold\_drinks

BHS School Bio Predator Free R...

**Burnside High School Canteen**

Search items here...

Hot Food (every day) Cold Drinks Sandwiches/Rolls/Wraps Muffins/Slices/Cookies/Cakes Potato Chips Salad (Summer only) Hot Food (Different Meal Each Day) Ice Creams Fruit & Yoghurt Hot Drinks

## Cold Drinks



Iced Coffee

Price: \$3.50



E2 Large

Price: \$4.50



Pump Water

Price: \$3.50

This is cold drinks url and title

When clicking on an item the details of the item should show on a separate page:



## Burnside High School Canteen

E2 Large



Price: \$4.5

Description: Large bottle of E2 sports drink.

Healthy Options: Gluten Free, Vegetarian, Vegan

[Back to Menu](#)

The url is also the id of the item:



127.0.0.1:5000/item/51

When clicking on the back to menu button or school logo it should direct back to the main page/menu page:



## Burnside High School Canteen

Welcome to the Burnside High School Canteen Menu!

[Hot Food \(every day\)](#)

[Cold Drinks](#)

[Sandwiches/Rolls/Wraps](#)

[Muffins/Slices/Cookies/Cakes](#)

[Potato Chips](#)

[Salad \(Summer only\)](#)

[Hot Food \(Different Meal Each Day\)](#)

[Ice Creams](#)

[Fruit & Yoghurt](#)

[Hot Drinks](#)



## Burnside High School Canteen

Search items here...

Hot Food (every day)

Cold Drinks

Sandwiches/Rolls/Wraps

Muffins/Slices/Cookies/Cakes

Potato Chips

Salad (Summer only)

Hot Food (Different Meal Each Day)

Ice Creams

Fruit & Yoghurt

Hot Drinks



Hash Brown

Price: \$1.00



Miso Soup

Price: \$2.50



Garlic Bread

Price: \$3.00



## Nielsen's Heuristic in my application:

### **User control and freedom:**

I added many home buttons and went back to menu buttons in my application so users can exit out of the page whenever they want.



**Burnside  
High School  
Te Kura o  
Waimairi-iri**

PRICE: \$2.5

Description: A warm, savoury Japanese soybean broth with a light, comforting flavour.

No healthy options available.

[Back to Menu](#)

### **Help and documentation:**

At the footer I added my contact information so if users run into problems or questions they can contact me so I can help them.

2025 Burnside High School Canteen, all rights reserved  
Contact 22011@burnside.school.nz for questions/errors

### **Recognize, Diagnose, and Recover from Errors:**

I added a 404 page for users with a link to go back to the main page if errors are made by users.

404 Not found, Please go back [Home](#)

### Match between System and the Real World:

In my database/code values/descriptions real humans should be able to understand what each category/section does and is. Because the wording is very simple and easy for people to understand.



## Burnside High School Canteen

Search items here...

Hot Food (every day)   Cold Drinks   Sandwiches/Rolls/Wraps   Muffins/Slices/Cookies/Cakes   Potato Chips   Salad (Summer only)   Hot Food (Different Meal Each Day)   Ice Creams   Fruit & Yoghurt   Hot Drinks



Hash Brown  
Price: \$1.00



Miso Soup  
Price: \$2.50



Garlic Bread  
Price: \$3.00



Vegan Pie (Kumera & Cashew)  
Price: \$6.5  
Description: Sweet potato and cashews in a plant-based filling.  
Healthy Options: Vegetarian,Vegan

[Back to Menu](#)

### Aesthetic and Minimalist Design:

Here I made every button/function as simple as possible. I also used less colours so people don't get confused/flashed by the application. Each button does what it does which is not too fancy to confuse users.



Burnside High School  
Te Kura o Waimairi-iri

**Burnside High School Canteen**

Search items here...

Hot Food (every day)   Cold Drinks   Sandwiches/Rolls/Wraps   Muffins/Slices/Cookies/Cakes   Potato Chips   Salad (Summer only)   Hot Food (Different Meal Each Day)   Ice Creams   Fruit & Yoghurt   Hot Drinks

DONE!