



Smart Scale



by bleuzetim

Hello everyone, I'm a student at Howest Kortrijk. For my first project I will make a smart scale. This because I'm really into sports and because I pay attention to my nutrition.

I know there are lot's of apps out there to track your food, but none of them are directly connected to a scale. So that's why I went and made one. Why you may ask. Because it's a lot easier to attach it directly to a scale, this way you don't have to weight it and calculate it on your own to add to the app or graph.

Supplies:

Raspberry pi:

- raspberry Pi 4 (2 GB model - 59.95€)
- 3A USB-C Voedingsadapter voor Raspberry Pi 4 (10.95€)
- SanDisk Extreme PRO SDHC-kaart 32 GB (15.90€)
- raspberry Pi t-piece (9.92€)

Sensores and modules

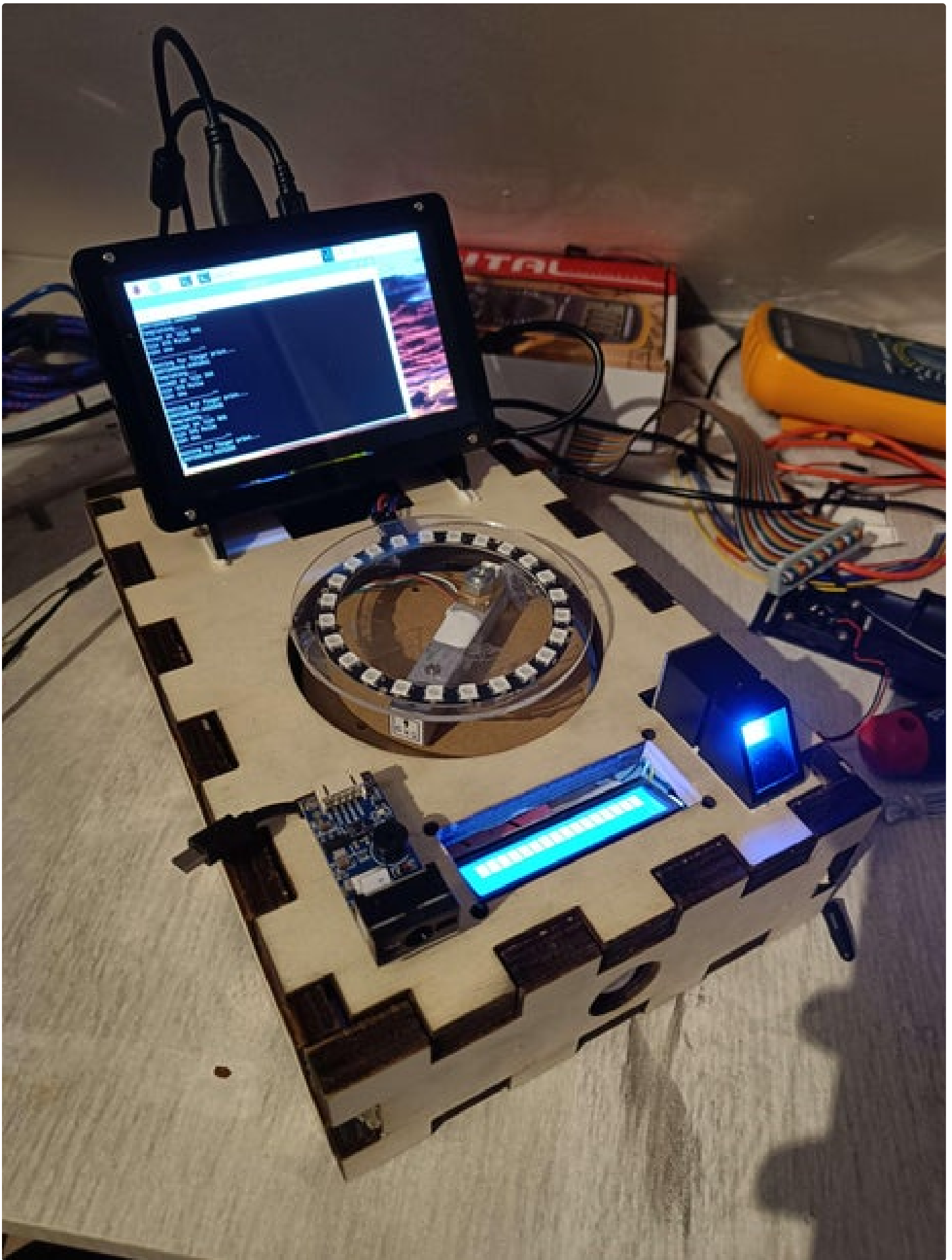
- DollaTek Optical Fingerprint Reader Module (17.99€)
- Waveshare Barcode Scanner Module (37.38€)
- Youmile HX711 Weight Sensor Module Kit Digital Load Cell Weight Sensor (15.99€)
- AZDelivery 5 x HC-SR501 PIR Motion Detection Module (6.99€)
- Neopixel 24 led ring (16.95€)

Other electronics

- AZDelivery 3 x HD44780 1602 LCD Display Module 2 x 16 characters (5.99€)
- 5 Inch Capacitive Touch Screen display (54.37€)
- breadboard + cables + breadboard power supply (optional - 13.99€)
- pcf8574 (2.44€)
- level shifter (3.50€)

For the box

- multiplex 12mm 600x450



Step 1: Setup the Raspberry PI

The first setup on the PI

The first thing we need to do is choose an OS. I chose the Raspberry PI OS and I also recommend you do the same, because this is the best for this particular job.

First off you should install the Pi imager (<https://www.raspberrypi.com/software/>). Now we have the software, now we need to install the OS image, go to this website : <https://www.raspberrypi.com/software/operating-systems/> and choose the 64-bit with desktop OS. After this we need to start the program and choose the OS. So we insert the sd card into our computer and start the pi Imager software. We go through the steps selecting our sd card and choosing our freshly installed os and wait till the program is finished.

Once the program is finished we still have to make some changes in the boot file. So open up file explorer and go to the boot drive. Here we have to choose for the file "cmdline.txt". Here we have to add the following piece of text to the bottom of the file.

```
ip=192.168.168.169
```

After this you save and exit the file. Then as the last step you have to make a ssh file in the same folder without any extension. If windows or ios gives it an extension automatically, then you have to delete it and save it.

Now it is ready to use and we can insert it in the Pi

Starting from the Pi

Now, in order to connect via ethernet to your pi, we have to connect a cable between our pi and pc. After this we have to put our ethernet address from our pc to statis with the following ip and subnet. The number placed on x you can choose, just not make it the same as the Pi.

```
ip: 192.168.168.x  
subnet: 255.255.255.0
```

Now that that is set up, we have to connect via SSH with our Pi, you can choose your favorite program for this. The ones I used were VNC viewer (<https://www.realvnc.com/en/connect/download/viewer/>) and putty(<https://www.putty.org/>).

Now open the program and connect to the right ip (192.168.168.169) with the right port (22). If you are asked to log in the default user is: "pi" and password: "raspberrry".

Settings

Now that we are connected we can start to set up the Pi as we like. Before we go further its best to keep the Pi up to date and run the following commands.

```
sudo apt update  
sudo apt upgrade
```

After this its best to reboot the Pi.

Once the Pi is rebooted we can go into the raspi config and change some things in there.

```
sudo raspi-config
--> Interface
--> I2C
--> enable I2C

--after reboot
sudo raspi-config
--> Interface
--> Serial Port
--> enable Serial
```

After this we have to check if we are in the Right timezone, so go to Localization and change the timezone and wlan country to the right country you're in.

After this we can reboot.

Wifi setup

Now we have to connect our pi to the internet, we can do this via the terminal as followed:

```
sudo wpa_passphrase 'Networkname' 'Password' >> /etc/wpa_supplicant wpa_supplicant.conf
wpa_cli -i wlan0 reconfigure
```

after this we have to reboot again.

Install Apache

In this and the upcoming parts we will download the necessary packages.

```
sudo apt install apache2 -y
sudo apt install mariadb-server mariadb-client -y
install python3
```

Clone Github

```
git clone https://github.com/howest-mct/2022-2023-projectone-bleuzetim
```

Apachte setup

If you check the apache site you should see the default apache 2 debian page. We will change this into our own site.

Database setup

Now we will setup the database and the user we will use I this project. To begin, we will run the following command.

```
sudo mysql_secure_installation
```

Just enter your password and choose all yes.

No we will go on to creating a user.

```
sudo mysql -u root -p
create user 'USERNAME'@'localhost' identified by 'USERNAME';
grant all privileges on *.* to USERNAME@localhost;
flush privileges;
exit;
```

Apache2

If you google the Ip address of your Pi you should see the default web page of apache2, if you don't it means it wasn't installed properly and you should try installing it again.

Now if you see the page, we will now change it to the right website by using the following commands. Start by opening up the .conf.

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

Find the line where it says "DocumentRoot" and change the path to your website.

```
DocumentRoot <your site path>
```

Now save the file using "ctrl+x" and click "y", after this we need to reboot our apache2 by doing the following.

```
sudo service apache2 restart
```

No after restarting we have to change one last thing. Start y running the following command.

```
sudo nano /etc/apache2/apache2.conf
```

And now scroll down until you see the following block of code.

```
<Directory />  
Options FollowSymLinks  
AllowOverride All  
Require all denied  
</Directory>
```

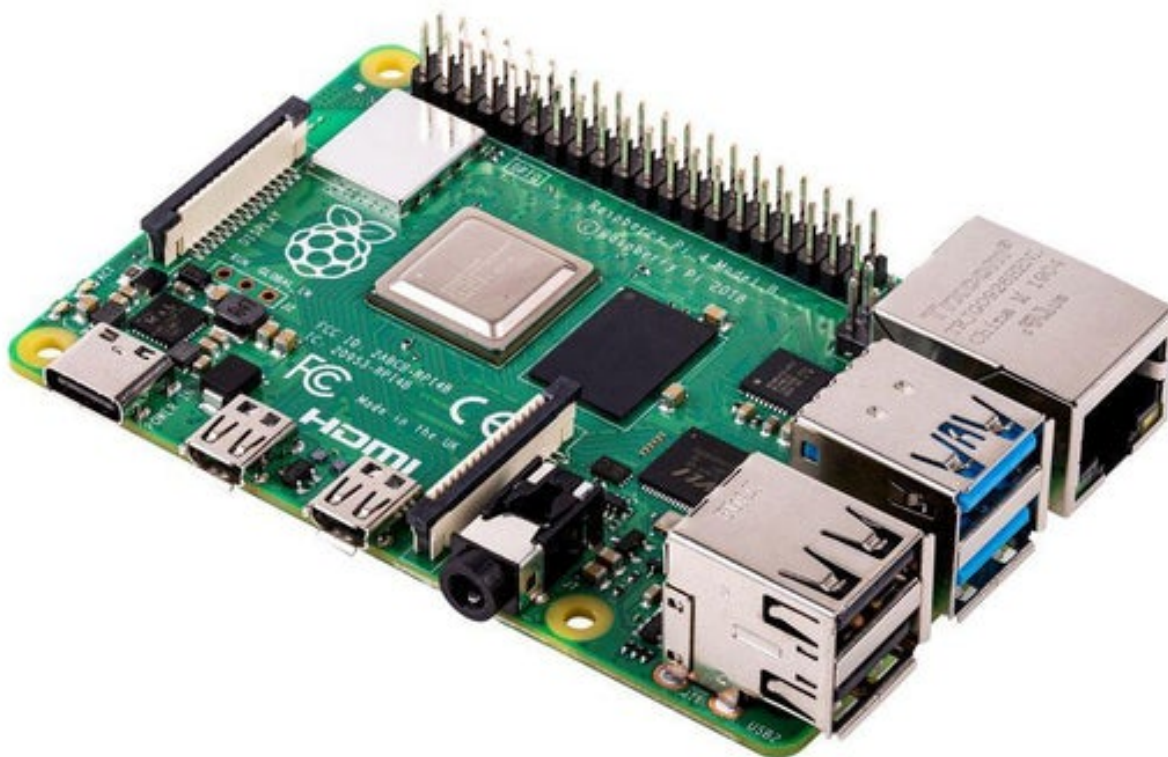
And change it into:

```
<Directory />  
Options Indexes FollowSymLinks Includes ExecCGI  
AllowOverride All  
Require all granted  
</Directory>
```

Safe the file and restart the service using the same command as earlier.



If you go to the site you should see your website now.

Adding the program as a service



Step 2: Fritzing

Pay close attention because orange=3.3V, red=5V and black=GND

	https://www.instructables.com/F1J/Z06J/LJ2V9WDR/F1JZ06JLJ2V9WDR.pdf	Download
	https://www.instructables.com/FYP/ZDMR/LJ2V9WDX/FYPZDMRLJ2V9WDX.pdf	Download

Step 3: Database

tblDevices

In this table we keep the devices we use stored.

tblHistory

Here we keep a history of when what sensor is used.

tblAction

This holds the actions we can do with our sensors e.x weight something.

tblCategorie

In this table we keep all our categories like flesh, fish,...

tblFood

Here we keep all our food with all the nutrition values we need.

tblMeal_has_tblFood

Here we keep the food connected to a certain meal.

tblMeal

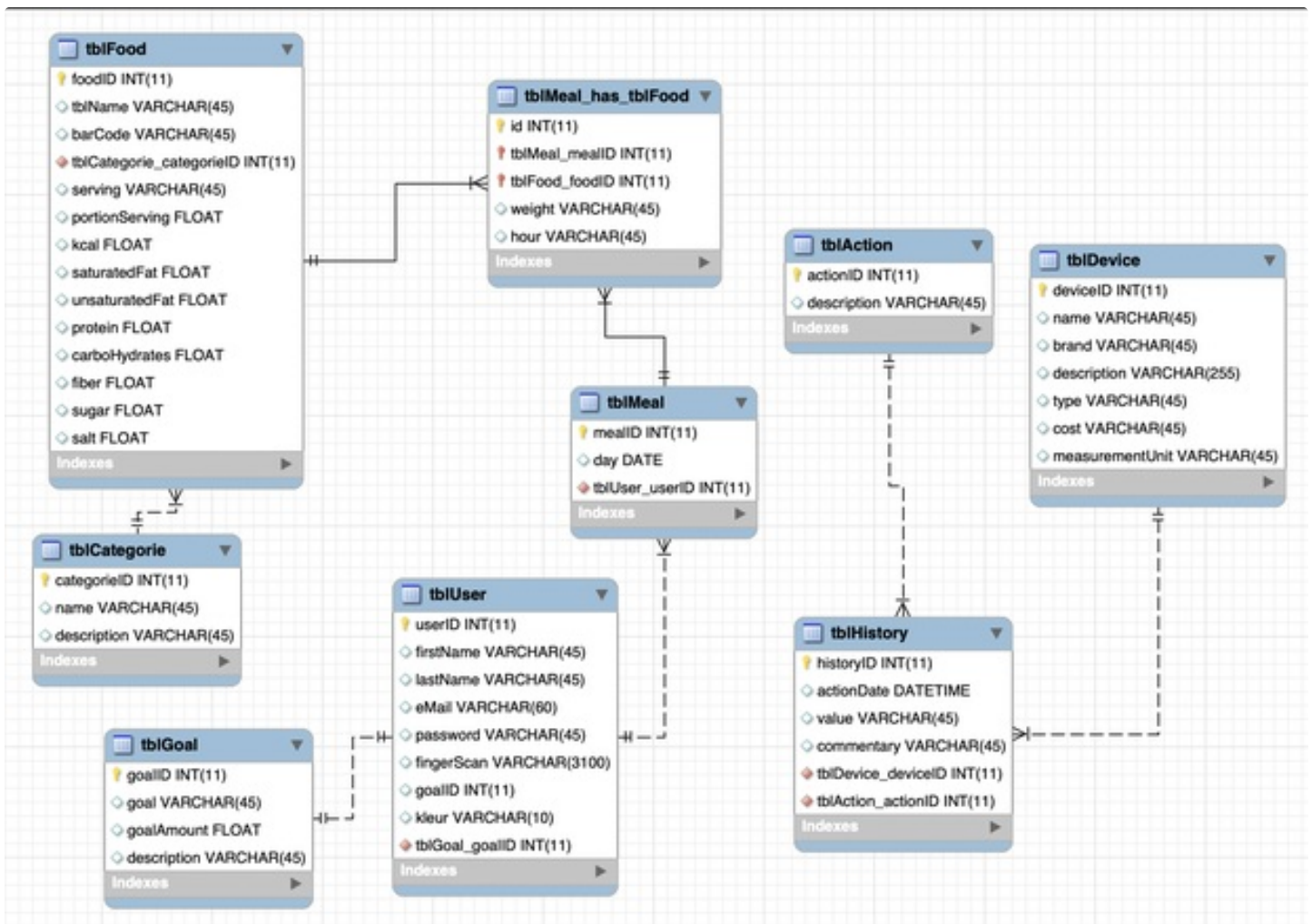
Here we keep all the meals with the date.

tblUser

Here we keep the data of the user, this is also linked with the meals table.

tblGoal

Here we keep the user his daily goal stored.



Step 4: The Backend

The entire backend is written using python(flask).

We use two different kind of ways to communicate with the front end.

1) via routes

We use routes for most of the updates and for some of the gets.

2) socketio

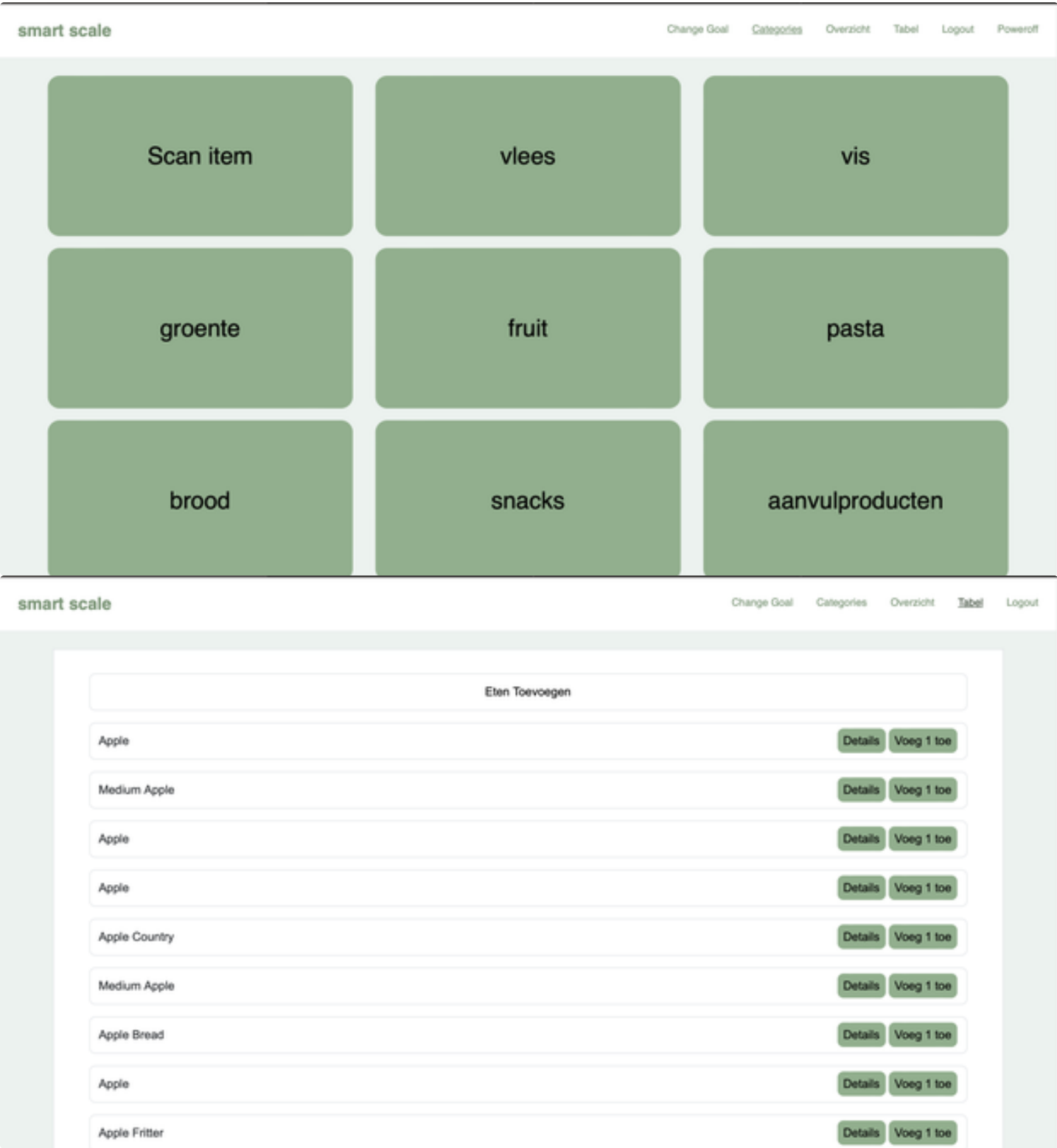
Then for the rest we use socketio, I like using socketio because it's easier than routes and we can use live data with socketio

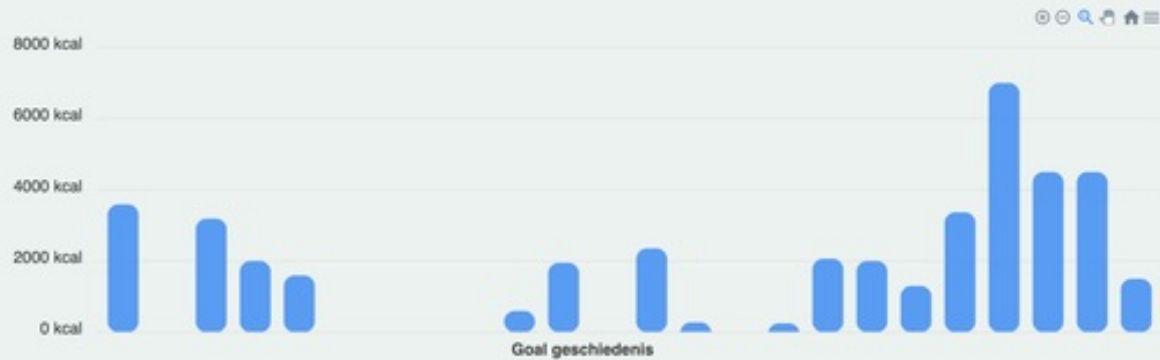
Now one last note, you should go to the config_example.py. Copy this file and rename it after config.py. Once this has happened you can fill in the credentials of the user we made earlier for the database.

Step 5: Frontend

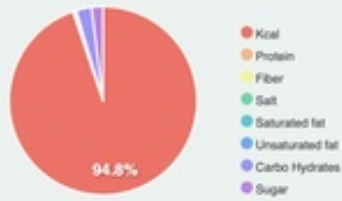
The frontend is all made with custom SCSS, html and a lot of javascript.

I brainstormed it in adobe Xd but didn't fully make it, because it would cost too much time.





Taart grafiek van de dag



Daily Goal:

kcal

Amount:

6000

Kleur:



Voeg Toe

Categorie

Vlees



Naam

Kcal:

Protein:

Carbo Hydrates:

Fiber:

Unsaturated Fat:

Saturated Fat:

Sugar:

Smart Scale

Klik hier om te registreren

LOGIN

Email:

Login

Smart Scale Register

Naam

Voornaam

Email:

Daily Goal:

Kcal

Amount:

Voeg Toe

Step 6: The Case

I decided to go with lasercutting because this was a cheap and fast way to make a cool looking case. Looking at it afterwards it was a good choice but I should've chosen a thinner kind of wood. I went with 12mm multiplex and this is way too thick. I recommend going with 6mm, but if you go with 6mm you can't use the image above as print image.

When it comes out of the laser cutter, you should still cut away some wood for the LCD display because the hole for me was a bit too small. But after this you can just use some wood glue and glue the bottom piece and 4 side pieces together. I recommend leaving the top open so you can still open the case in need of emergency's or when you want to change some electronics later.

Than use the circle you got from the lasercutting as a little step up to glue under the weight sensor so it pops out of the case like seen on the pictures.

