

eYSIP-2017

# DEVELOPMENT OF WEB INTERFACE FOR GH FARM PRODUCE



Hemang Gandhi

Keivan Shah

Mentors : Yogita Mali, Uma Sharma, Parin Chedda

Duration of Internship: 22/05/2017 – 07/07/2017

*2017, e-Yantra Publication*

# Development of Web Interface for GH Farm produce

## Abstract

This project is part of the Eyantra Summer Internship Programme 2017. The main aim of this project to make a automated system for the Logging and Monitoring of Farm produce of greenhouses. The system consists of a smart weighing machine powered by Raspberry Pi which logs all the data about the produce such as weight, crop type, trough id, time, etc. The details are then sent to server which maintains all the data and can provide analytics about the data. The smart weighing machine was developed last year as part of eYSIP'16 and the main goal of this years project is to improve the weighing machine, setup a web interface for the data and if possible an e-commerce website to buy and sell the GH farm produce.

## Completion status

The project is successfully completed. We have successfully created an interface for each of the three users namely producer, consumer and admin providing different functionalities.

## Hardware Parts

- Hardware parts
  - Load cell
  - 20x4 LCD
  - 4x4 Keypad
  - Raspberry pi 2 Model B



## 1.2. SOFTWARE USED

---

- LM2596 step-down voltage regulator
- HX711(ADC amplifier)
- USB Camera
- DC Adapter
- Enclosure to hold all parts

- Detail of each hardware:

[HX711 ADC Amplifier](#)

[Load cell CZL601](#)

[Step down voltage regulator](#)

[20x4 LCD](#)

[Membrane Keypad](#)

[Raspberry pi 2 Model B](#)

[4x4 Keypad](#)

Hardware is not in the scope of this project Please refer last year's [repository](#) to get more details about the weighing machine and connections

## Software Used

- Python 3.5.2
- Django 1.11.1
- Bootstrap 4
- HTML 5

For installation of required softwares refer our [wiki](#)



## Software and Code

### Smart Weighing Machine

The smart weighing machine consists of a Raspberry Pi, LCD display, keypad, load cell and a camera. To log the produce the producer places the produce from the GH farm on the weighing machine. The weighing machine with the help of the the load cell displays the weight on the LCD screen. So the machine can be used as a normal weighing machine. It also has standard weighing machine features like tare. Now the producer can log the produce by pressing \* on the keypad. The Raspberry Pi then takes a picture of the produce and sends it to a prediction server to guess the type of the crop. The producer is then shown the possible crops as predicted by the Vegetable Identification using Transfer Learning project. If the crop predicted is correct, the producer can continue or else he can change the crop by selecting one from a list of crops shown sorted according to the confidence of the predicting system or by inputting in the crop id. Once he enters the crop id the enter data is sent to the server for Logging. The data is logged and also made available for selling on the e-commerce website. The data is sent via HTTP requests making use of the modern REST API.

### Web Interface

The web interface for this project was written in Python using Django. The web interface has mainly three applications, they are:

#### **FarmApp:**

This is the main App of the system. It provides with all the UI that the users i.e. Admin, Producer and Customers interact with.

#### **Machine:**

This app is used only by smart weighing machine to log the produce. The data logged by the machine is sent via HTTP requests to this server for logging. Once the data reaches here it added into the system for buying as well as monitoring.

#### **Predict:**

This is an utility app that runs the Vegetable Identification algorithm on the image sent to it and returns a list of possible crops along with the confidence



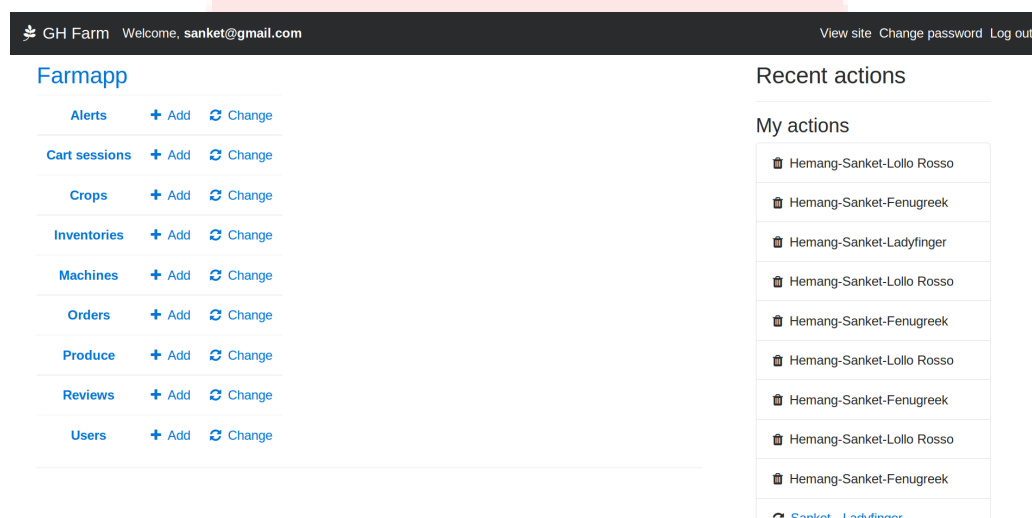
## 1.4. USE AND DEMO

percentages. This app is used by the smart weighing machine to guess the vegetable by the picture it takes with the webcam attached to it.

## Use and Demo

### Admin Interface:

Admin has control over the entire database and make modifications to the data as and when required. It is his responsibility to add new Producers or Crops into the database.

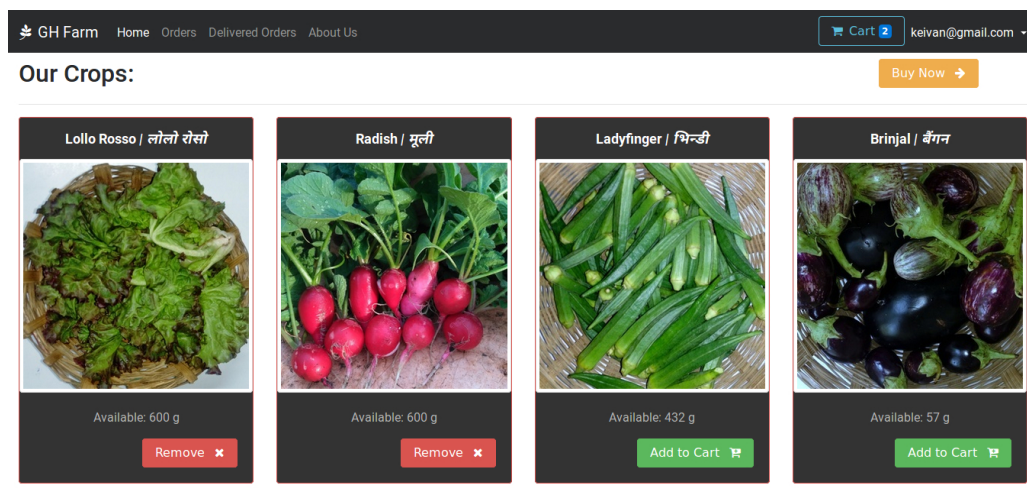


### Consumer Interface:

Consumers are the end users of the entire system. They can register onto the system and can buy the produce from the system and also rate the producers.

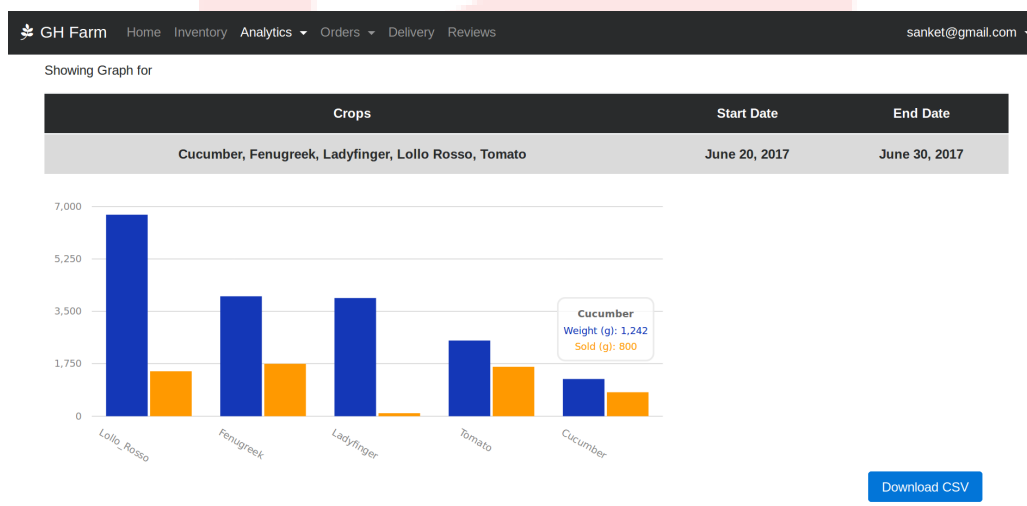


## 1.4. USE AND DEMO



### Producer Interface:

Producers are the users who own the smart weighing machine and use them to add their produce to the system for monitoring and also to sell them. Each producer can have multiple machines allotted to him and all produce by this machines is logged into his account.



Instructions to use the website can be obtained in our [wiki](#).  
Visit the website : [store.k-yantra.org](http://store.k-yantra.org).



### Future Work

**1. Include bill payment options:**

Currently the website does not include a billing system which could be added in future.

**2. Improve design of web interface:**

The current UI is simplistic. It can be made more user friendly by improving the design.

### Challenges Faced

**1. Handling Race Condition:**

When a number of users concurrently access and modify the database it might leave the database in an inconsistent state. In order to avoid this situation we made use of transactions to maintain atomicity and made use of lock based protocol to establish concurrency control between the transactions. In lock based protocol, at a given time only one transaction can modify a table cell. This ensures consistency of the database.

**2. Caching:**

One of a problem which we had never seen before was caching. The problem was that whenever the user filled up a form and was directed to a new page he can return to the filled up form by going back on the browser even though the server code did not allow it. This was because of local caching by the browser. We later found a solution to this problem by including do not cache headers into HTTP response from the server.

**3. Offline storage of data on raspberry pi:**

There may occur a situation in which data sent by the smart weighing machine to the server is not sent properly due to no internet connectivity or some other errors.

In such a situation data loss would occur. Thus the server sends an acknowledgement when it receives data from the smart weighing machine. In case of a failure to send data the machine would not receive appropriate acknowledgement and would store the data offline and try again later. This ensures that no data loss takes place.

# Bibliography

- [1] Django Documentation : <https://docs.djangoproject.com/en/1.11/>,
- [2] Djangogirls Tutorial : <https://tutorial.djangogirls.org/en/>,
- [3] Bootstrap 4 : <https://v4-alpha.getbootstrap.com/>