Dustbin IoT

2020

“Superior Technician for Smart City & Clean Energy Management”

Ariel Montes Nogueira • info@montesariel.com

INDICE

Dustbin IoT 1

Description 1

Utilization 1

Technical features 1

The application 1

The Interoperability 5

Other information 6

Project Analysis 7

Bibliography 8

DUSTBIN IOT

Description

Intelligent system for managing the refuse collection of a Smart City.

It’s a network of "sensor nodes" (i.e. devices that incorporate a detection sensor) positioned under each city waste bin that communicate on the network through special gateways, providing real-time data to a server. The data are also stored in a relational database with different levels of authorization and can be consulted H24 via a web interface. and updates to this web app can be performed without interrupting the service. Remote control of the aforementioned nodes is also planned.

Utilization

The application is accessible online **[1]** 24/7 and updates can be performed without interrupting the service. The administrator can add, modify or delete from the application the instances that represent: orders, containers, operators and accesses. Each user sees their orders and the capacity level of the waste containers to be collected. Expired orders will not be viewed by operators.

Technical features

Scalable web application (allows you to add new features to the program). Written in Python / Django and distributed on the Heroku platform, it uses PostgreSQL as a database using the Django ORM. And JavaScript, HTML, CSS for the Frontend.

The application

Below, I explain in detail the functionality of the web application using the images taken from it.

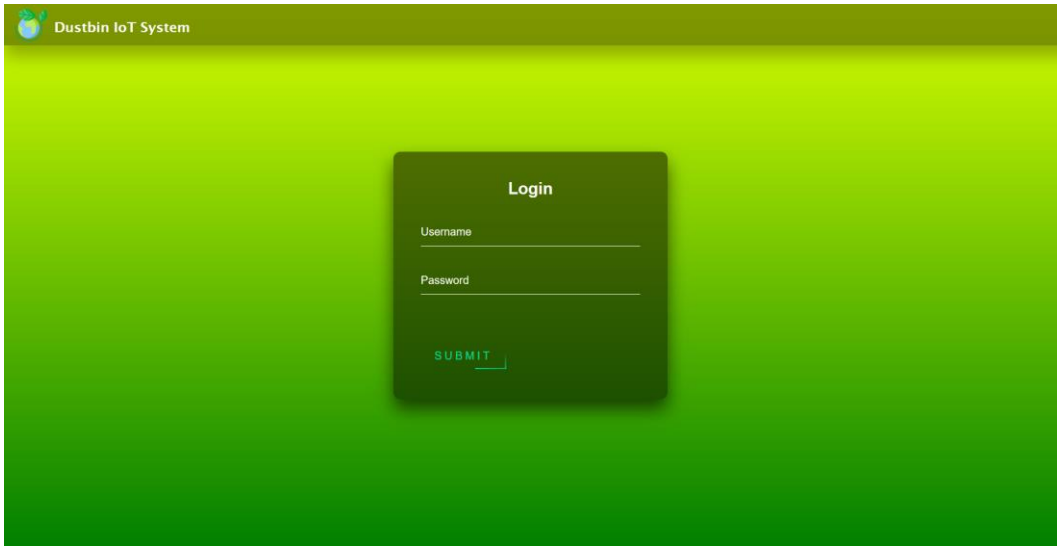

Figure 1 (System login page)

From figure 1 it is possible to see how access on this page is managed as operator or administrator. What will be displayed on the next screen depends on your access privileges.

Login passwords are automatically encrypted before being saved to the database.

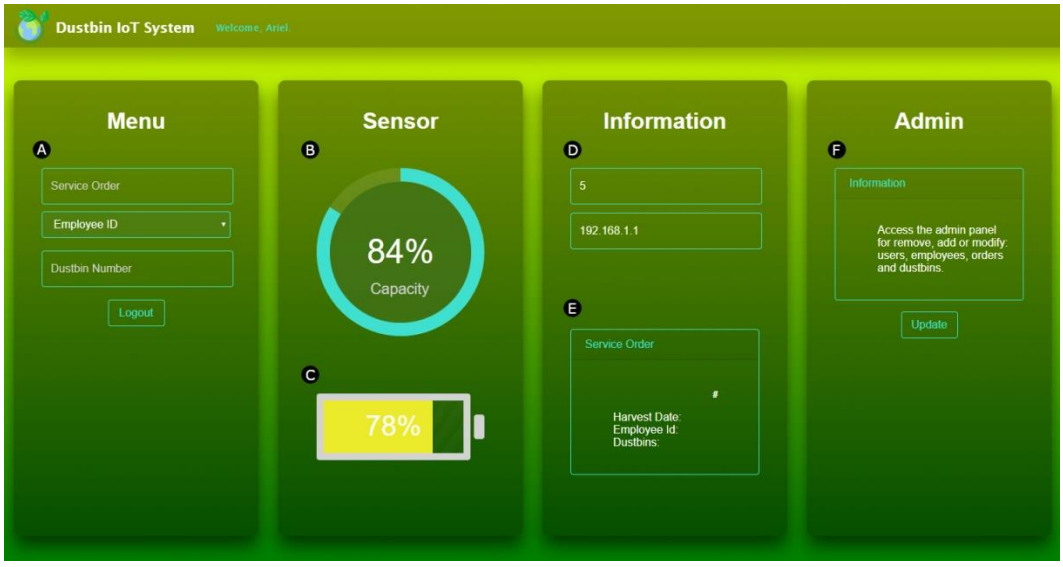

Figure 2 (System interface of the administrator)

Once logged in as an administrator, the interface shown in figure 2 will be displayed, containing the points listed below.

- In point A, it is possible to display a form to obtain sensor information by filtering it by ID, access all service orders based on their identification number, check the service orders closest to the deadline for each operator and exit the session with the logout key.

- In points B and C, we find a graphic representation of the sensor data, i.e. the capacity of the bin and the state of the battery that powers the sensor.

- The fields in point D are used to show the identification number (or ID) and the IP address of the sensor. The latter can be useful for interventions in the configuration of the network.

- Point E is used to display information on the service order such as the order number, the pick-up date, the operator code assigned and the list of baskets to be emptied.

- Point F allows access to the control panel where it is possible to carry out critical impact activities on the program, such as denying access to all other users. For this reason only the system administrator can access it.

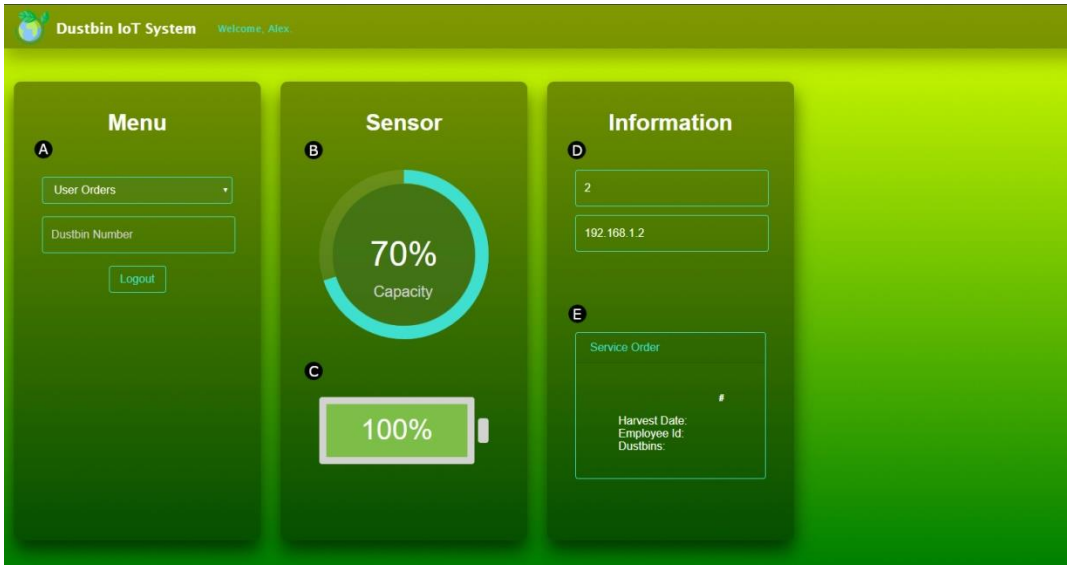

Figure 3 (System operator interface)

The differences of the administrator interface (figure 2) compared to those of the operator interface (figure 3) are listed below.

- The first thing to note in figure 3 is the absence of the administrator menu (figure 2, point F) to prevent access to the control panel.

- In point A of the operator interface (figure 3) it is not possible to access all service orders, but only those assigned to the operator who has logged in, he is also allowed to view information relating to the sensors.

The functions of the rest of the points do not change.

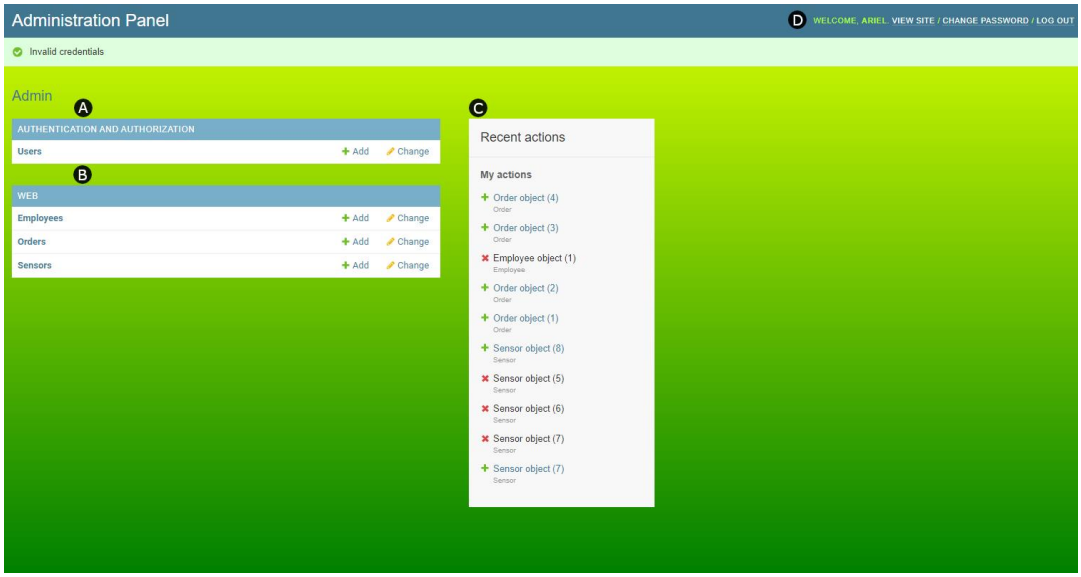

Figure 4 (Administrator control panel)

Inside figure 4:

- Point A allows the administrator to modify or create new accesses (or users) in the application.
- In point B, it is possible to create or modify new orders, sensors or operators. However, the latter are automatically generated after creating a new access to the application.
- At point C, the latest changes recorded within the control panel are simply displayed.
- Point D provides the administrator with the options to return to the home page, change the login password and log out.

The interoperability

The data collected by the sensors of the waste bins must be delivered to the application database in order to subsequently allow their visualization.

The sending of these data takes place via a computer (PC, Raspberry or Arduino) with internet access and dedicated to the recopilation, processing and sending of the final information in JSON format to the database. Another program that I developed (hereinafter referred to as "db-filler") is responsible for all of this. It

allows you to carry out these activities without compromising data integrity or database security.

In the security field, in addition to error management, it allows access to the database by means of a password which is requested at the first access or if the computer is restarted.

The following line of code inside the db-filler obtains the URL which allows access to the database and which requires the password in case you are not authenticated.

```
DATABASE_URL = popen('heroku config:get DATABASE_URL -a dustbin-iot').read()[:-1]
```

Before writing the new data, the db-filler performs checks in the database using queries, as can be opened in the following code.

```
# Create new sensors if id is None
if i['id'] is None:
        cur.execute("INSERT INTO web_sensor (capacity, battery, ip, location) VALUES(%(capacity)s, %(battery)s, %(ip)s, %(location)s)", i)

else:
    # Create new sensors with a specified id
    if not exist(i):
        cur.execute("INSERT INTO web_sensor (capacity, battery, ip, location, id) VALUES(%(capacity)s, %(battery)s, %(ip)s, %(location)s, %(id)s)", i)

# Update sensors (find sensor by id)
cur.executemany("UPDATE web_sensor SET capacity=%(capacity)s, battery=%(battery)s, ip=%(ip)s, location=%(location)s WHERE id=%(id)s", sensors)
```

These checks allow you to identify if a sensor is new and a new instance must be created or if it already exists and needs to be updated.

Furthermore, it is possible to synchronize the execution of this program with the moment when the data from the sensors will be received for greater performance optimization.

Other information

Before developing the web application, I made a conceptual prototype using the Node-Red framework.

Further information on the application and its prototype, such as the code or the informative video, are available online **[2]**.

PROJECT ANALYSIS

As explained above, the Dustbin IoT project is scalable and allows the integration of new features. This allows this web application to progressively improve. Below is a list of some of the features that could be implemented to improve this program.

- Automatic sending of emails or notifications to operators for communicates the assignment of a collection.

- Auto generation of service orders based on following parameters: operators with less workload, capacity of the baskets and suitable days for collection.

- Calculation of the most efficient route to perform the collection e possibility to display it on a map with access to GPS from the application.

- Support in multiple languages for more understanding from part of the operators.

- Tutorial on how to use the application at first access, repeatable if necessary.

- Possibility to identify who is the responsible for the current day and if necessary contact him from the app.

BIBLIOGRAPHY

[1] A. Montes, «Dustbin IoT» [Online]. Available in: dustbin-iot.herokuapp.com

[2] A. Montes, «Project 1» [Online]. Available in: montesariel.com/portfolio/project-1