

# INVENTORY MANAGEMENT SYSTEM

## Introduction

Inventory management system is a web-based application built using the Django framework for managing inventory. This manual provides an overview of the features and functionality of the application, as well as step-by-step instructions for common tasks and installation.

### Libraries used:

- Bootstrap - For the frontend styling, it can be further modified to be more responsive. It has been designed to be used on a PC screen.
- Django-jazzmin - For a better user interface in the admin panel
- Reportlab - to export the information in pdf.

## Operating mode

It is only possible to access the application if you are registered. Within the application there are two types of users with different permissions.

### - Administrator

The superuser is the administrator of all inventories and is also solely responsible for the creation of items and allocation of the total amount of each one.

When a super user creates an inventory for a department, a regular user with the same name is automatically created.

For example, when creating the inventory of the IT department, the user "it" with the password "it\_pass" will be created.

The administrator is able to access the administration panel and change the passwords for added security.

The application has two forms of view. One for the super user, who can access the general inventory, create, edit, delete and modify the quantity of each item or its category.

You can also access the inventories of each department, create, edit, delete or modify them.

### - Regular User

The regular user can only access his own inventory (he won't see the item overview, he won't be able to add items to the general inventory or see other departments' inventories).

He will be able to add the available items from those previously added by the administrator and modify the amount he owns in his department.

## Database modeling

The basic models are Inventory (assigned to a user, including the 'department' field), Item (with its total quantity, category, available quantity) and ItemInventory (intermediary model to manage the relationship between inventories and items).

The item has a field called "total" and a field called "available". When a user or the administrator assigns a certain amount of items to some inventory, the value of available items decreases automatically. At the same time, it will not be possible to assign more items to an inventory than are available.

### -- Export and search

The application allows you to export the filtered data, by name or category, of each inventory or the general inventory in pdf format.

### -- Dynamic content

Thanks to the Django architecture, the application has been developed dynamically, in such a way that a single template is used to display the information according to the search, the type of user and different conditionals depending on each need.

# INSTALLATION (for Windows users)

## Prerequisites

- In order to make it run on a local server, it is necessary to have python installed.  
(official download page for windows  
<https://www.python.org/ftp/python/3.11.2/python-3.11.2-amd64.exe> ) and  
make sure to accept the inclusion in PATH.
- It is also useful to have pip installed. (Here you can find a guide for this:  
<https://pip.pypa.io/en/stable/installation/> )
- And Git ( <http://git-scm.com/download/win> )

## GET READY

1. In a console or code editor, open the folder where the project will be located.
2. Create a virtual environment with the command:  

```
python -m venv env
```
3. Activate the virtual environment  

```
source env/Scripts/activate
```
4. Clone the code  

```
git clone '#'
```
5. Still from terminal or code editor, access into the directory where manage.py is located and install the necessary packages.  

```
pip install -r requirements.txt
```
6. Start the database migrations:  

```
python manage.py makemigrations  
python manage.py migrate
```

7. Create a superuser:

```
python manage.py createsuperuser
```

8. Power on the server:

```
python manage.py runserver
```

( # here you can decide which port you want to run the Django server on. The default port is localhost port 8000. However to change this you can write for example: `python manage.py runserver 127.0.0.1:80` #)

9. Access from an internet browser (such as google chrome or mozilla firefox) to `http://127.0.0.1:8000`

10. ALL READY! You can now use the inventory management application.

## FIRST STEPS

Remember that you are accessing with the superuser that you previously created, so you will have access to all departments and full control of the application. There are no inventory or items in the database yet, but you can start creating the ones you want.

Keep in mind that when creating an inventory for a department, you will automatically be creating a user associated with that inventory, who will only have access to its items and inventory.

You will have to create some items. As administrator you must to, the regular user cannot create items, he can only add the available items in his inventory.

## AUTHENTICATION AND SECURITY

The automatically created user will have the same name as the apartment (changing the spaces by \_ and the upper case by lower case) and a similar password ending in \_pass.

If, for example, there is a department called *Natural Sciences*: the username will be "natural\_sciences" and the password will be "natural\_sciences\_pass".

If we have a department called "IT", the username will be "it" and his password "it\_pass".

*# So don't forget to change the passwords ! #*

To change the password of the users, you can do it from the administration panel, whose link is in the navigation bar of the web app.

Just go to users, select the desired user and change his password.