**Inventory-Manager**

An inventory management system using Flask

**Getting Started**

**Clone this repository and set your path to it's folder, to get it up and running on your local system.**

git clone https://github.com/marination/Inventory-Manager.git

cd Inventory-Manager

**What to look for here?**

* [System Summary](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#system-summary)
* [Running the app](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#running-the-app)
* Features
  1. [Adding Products and Locations](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#adding-products-and-locations)
  2. [Deleting Products and Locations](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#deleting-products-and-locations)
  3. [Moving Products](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#moving-products)
  4. [Editing Products and Locations](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#editing-products-and-locations)
* [Built Using](https://github.com/marination/Inventory-Manager/tree/35e1986ce6349fba023cbf3080e869f6da463d7e#built-using)

**Prerequisites**

To run this system you will need :

* Python 3
* Flask
* SQLALCHEMY
* WTForms

Assuming you have Python, proceed to install the rest using the command below:

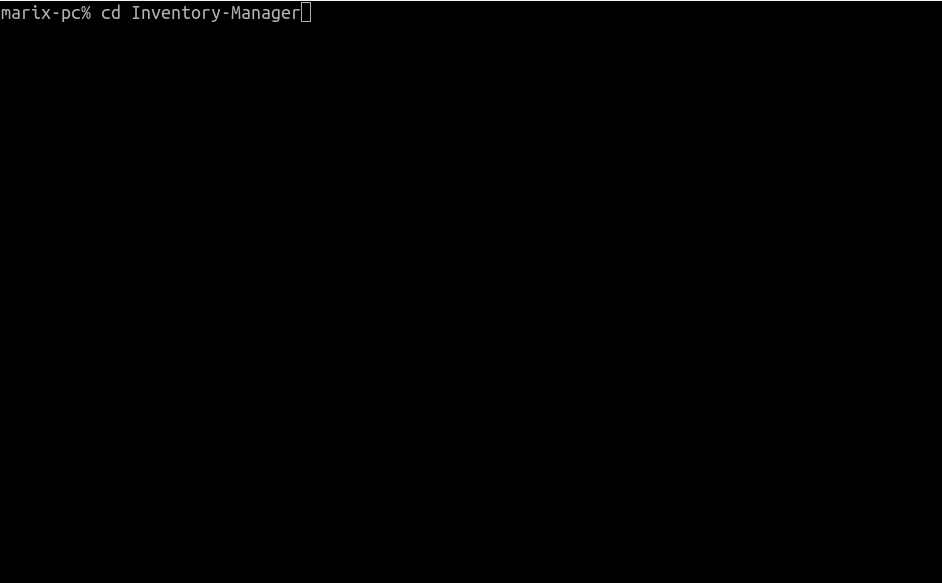
pip3 install -r requirements.txt

**System Summary**

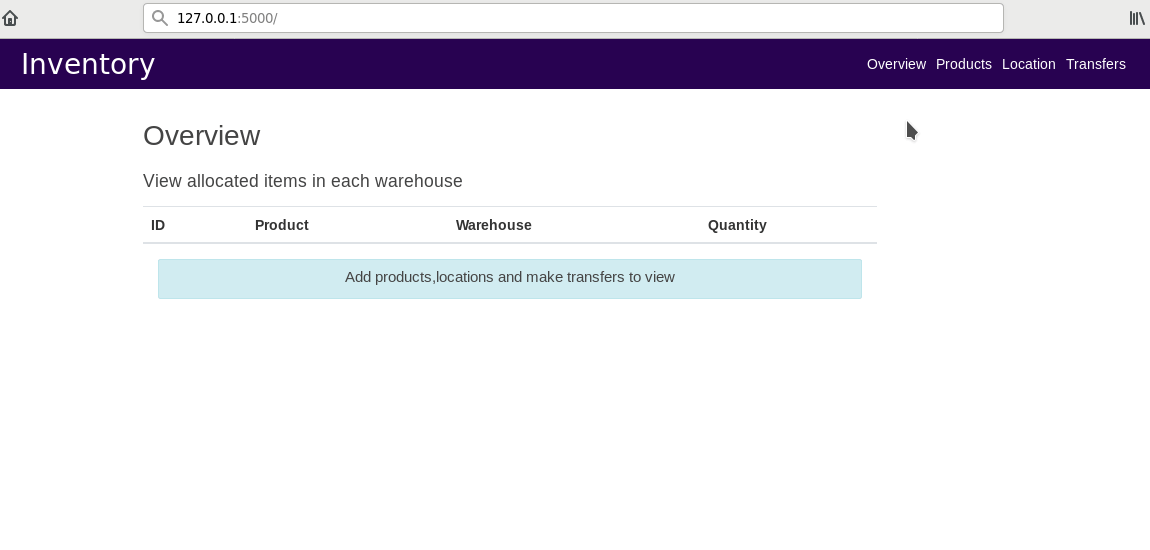
This system is built to simulate a warehouse environment and handles balancing quantities over warehouses. It has 4 main views including *Overview*,*Products*,*Locations* and *Transfers*. **Products** and **Locations** let you add,edit and delete entries from the system. **Transfers** lets you move items into the central warehouse, out of the central warehouse; also to and from various locations.It also displays transfer history. **Overview** will display products,warehouses and their respective balanced quantities.

**Running the app**

1. Set your current path to where the cloned folder is and run the file **run.py**

[](https://user-images.githubusercontent.com/25857446/56443542-c4926380-6312-11e9-98ac-42aa6830bf42.gif)

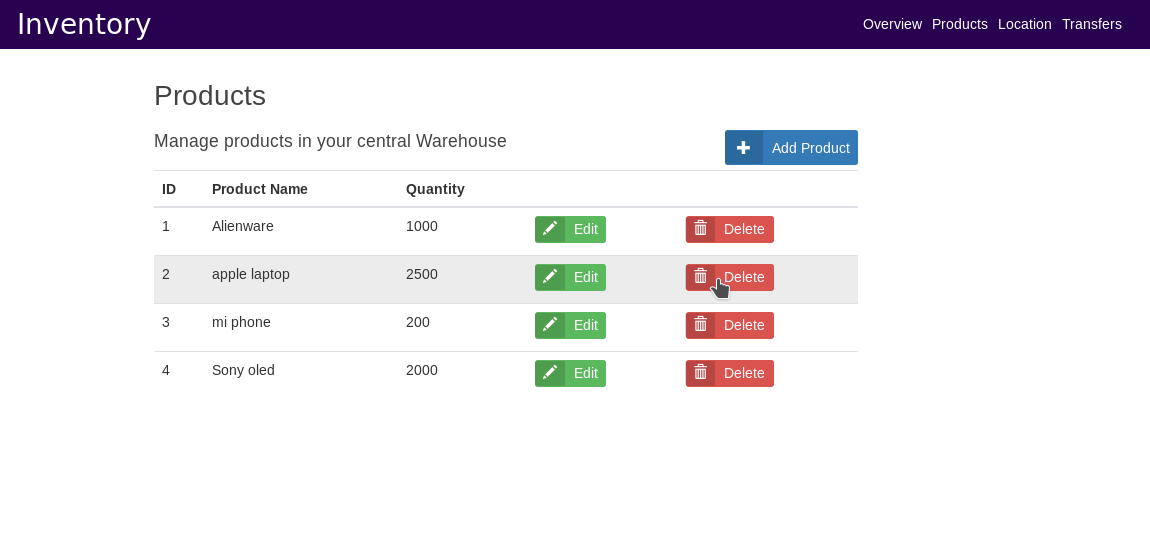
1. Either copy paste the url as shown above into your browser **or** simply check into *localhost:5000/* as shown below. You will see the initial views of each page as no actions are performed.

[](https://user-images.githubusercontent.com/25857446/56443683-500bf480-6313-11e9-9397-4ec93a34d29d.gif)

**Features**

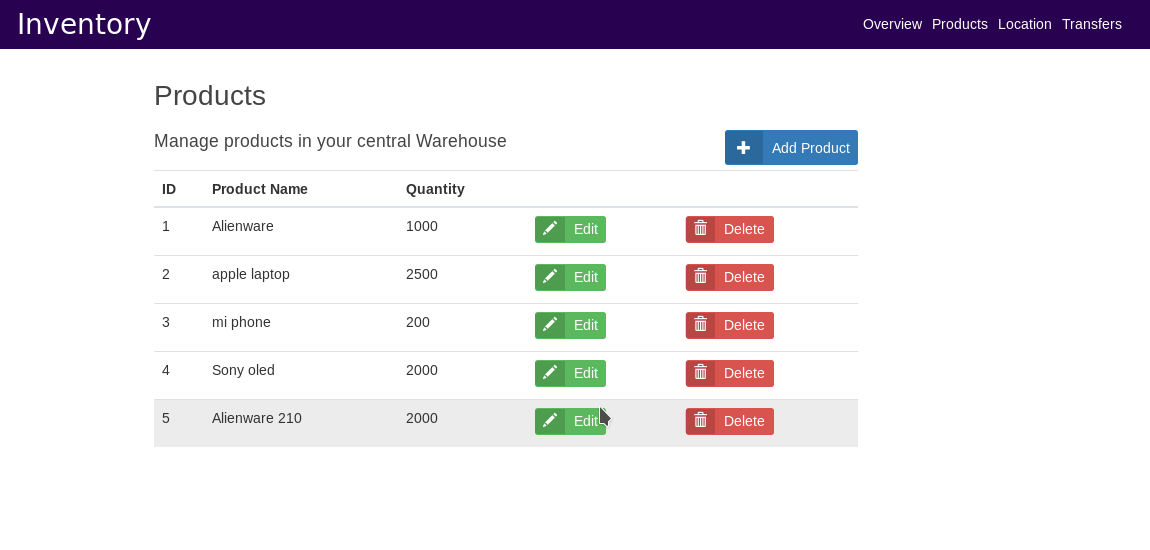
**Adding Products and Locations**

Products require product name and quantity to be filled. Location only requires location name

[](https://user-images.githubusercontent.com/25857446/56444083-e55bb880-6314-11e9-87de-8deabdc1c6a9.gif)

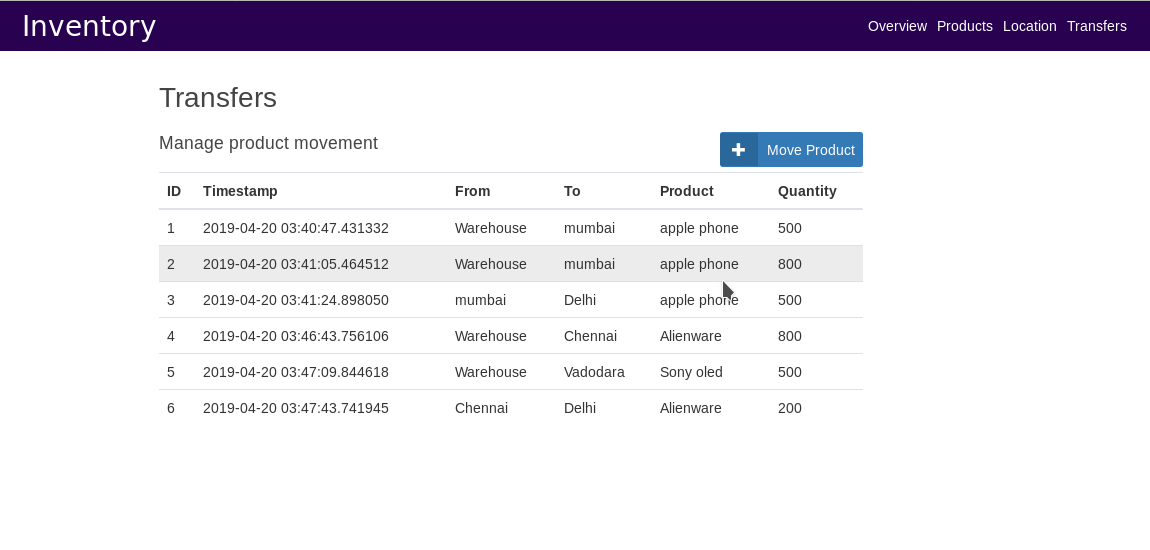
**Deleting Products and Locations**

Deleting only requires a button click, although the transfers(if any) will remain in the history.

[](https://user-images.githubusercontent.com/25857446/56444188-5307e480-6315-11e9-83d8-afaeda5d39ff.gif)

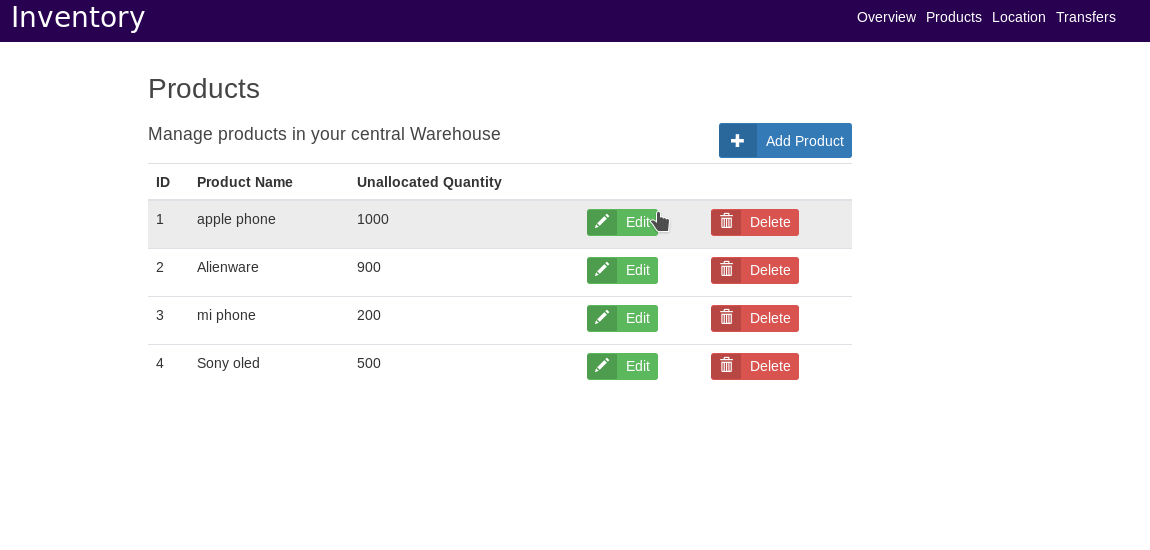
**Moving products**

Here products can be moved to a location, from a location as well as to and from a location. Products need to initially be added to various locations from the central warehouse.

[](https://user-images.githubusercontent.com/25857446/56446389-04137c80-6320-11e9-9c68-041db8b00a19.gif)

**Editing Products and Locations**

Change in product or loaction name creates changes in their names in the history and system overview.So, you can rectify a spelling error and still not loose any data.

[](https://user-images.githubusercontent.com/25857446/56446569-fb6f7600-6320-11e9-85e5-f67e6a454e26.gif)

**Built using**

* Flask
* SQLAlchemy