# Introduction

Our Herbco system design is quite simple but very functional. This web-site provides to buy the herbal products or medicines by the visitors. Customers can easily find what they want with our functional system design. After the design stage, we can decrease the time required the implementation.

## Purpose of the System

As we said in RAD, this project is a shopping site where users can access herbs and herbal products online in order to reach a healthier life or to get rid of their disease. The main aim of this project is to help the user to find herbs and herbal products that will be good for the health of the user, depending on the health condition or disease. The Herbco website, which contains several quality brands, contains products that the user is looking for. Distributors can add herbal products and herbs to the system and its information. This system also allows the user to place order which will add the items to the user’s cart and make payment for the same. As a result, this system helps the user to achieve a healthier life and cure his or her illness.

## Design Goals

The purpose of our project is to fulfill the requirements of software engineering completely. It is to specify all functional and nonfunctional functions together. With the definition of the functions, we have defined all the requirements for our Herbco project and have prepared an infrastructure for future versions or new projects. In our Herbco system design, we provide to our users or our visitors with easy access to our website to search for herbal products with product names or categories, and read product descriptions, registration and shopping. The features our system evaluates based on non-functional functions are as follows:

* **Dependability**

One of the most significant non-functional requirement is system security. The user security is on the front board in our Herbco system. In addition to security and safety, we paid attention to robustness, reliability, availability and fault tolerance criteria to make a complex system.

* **Maintenance**

Maintenance of your Herbco system is periodically performed by the administrator. Of course, when we are creating to our system, shortcomings such as extensibility, modifiability, adaptability, portability and readability were taken into consideration.

* **End User Criteria**

On our Herbco website, users and visitors can search, sign up, advance search, view their own information, and buy what they want by looking at their products.Our system efficiently stores and retrieves user data in a dynamic manner. In addition , we noticed that utility and usability factors are important for us. The Herbco system supports Microsoft and MacOS operating systems.

* **Performance**

Our Herbco system is responsive and it can accomplish a maximum number of tasks easily. The memory space of our system is available for speed optimizations. As we mentioned the response time, through put and memory criteria are significant for our system.

* **Cost**

We try to accomplish optimal level for cost of our system when we develop it. Also this cost not only for design considerations but managerial , as well. The maintaining backward compatibility with a previous system can add to the development cost while reducing the transition cost. By the way, we handle the development cost , deployment cost , upgrade costs , maintenance cost and administration cost.

## Definitions, Acronyms, and Abbreviations

The abbreviations and definitions contained in the document are given below:

* Herbco: Herb Shopping Company
* Admin: Herbco system admin
* Distributor: Company and brand CEO who sell their products on the site
* Manager: The manager and owner of this system
* Model: A schematic description of a system that accounts for its known or inferred properties.
* System: Any interacts by the application are considered to be done by the system.
* Efficiency: The properties of an algorithm, which is the amount of computational resources used by the algorithm.
* Service: Service is a keyword. Purpose of the service is to provide the customer with a secure payment system.
* OOP: Object Oriented Programming
* POP: Procedural or Produce Oriented Programming
* SQLite: Structed Query Language Lite
* HTML: Hypertext Markup Language
* CSS: Cascading Style Sheets
* MIT License: Massachusetts Institute of Technology License
* API (Django): Application Programming Interface
* SDD: System Design Document.

## References

Requirements Analysis Document (17.10.2017)

<https://www.djangoproject.com>

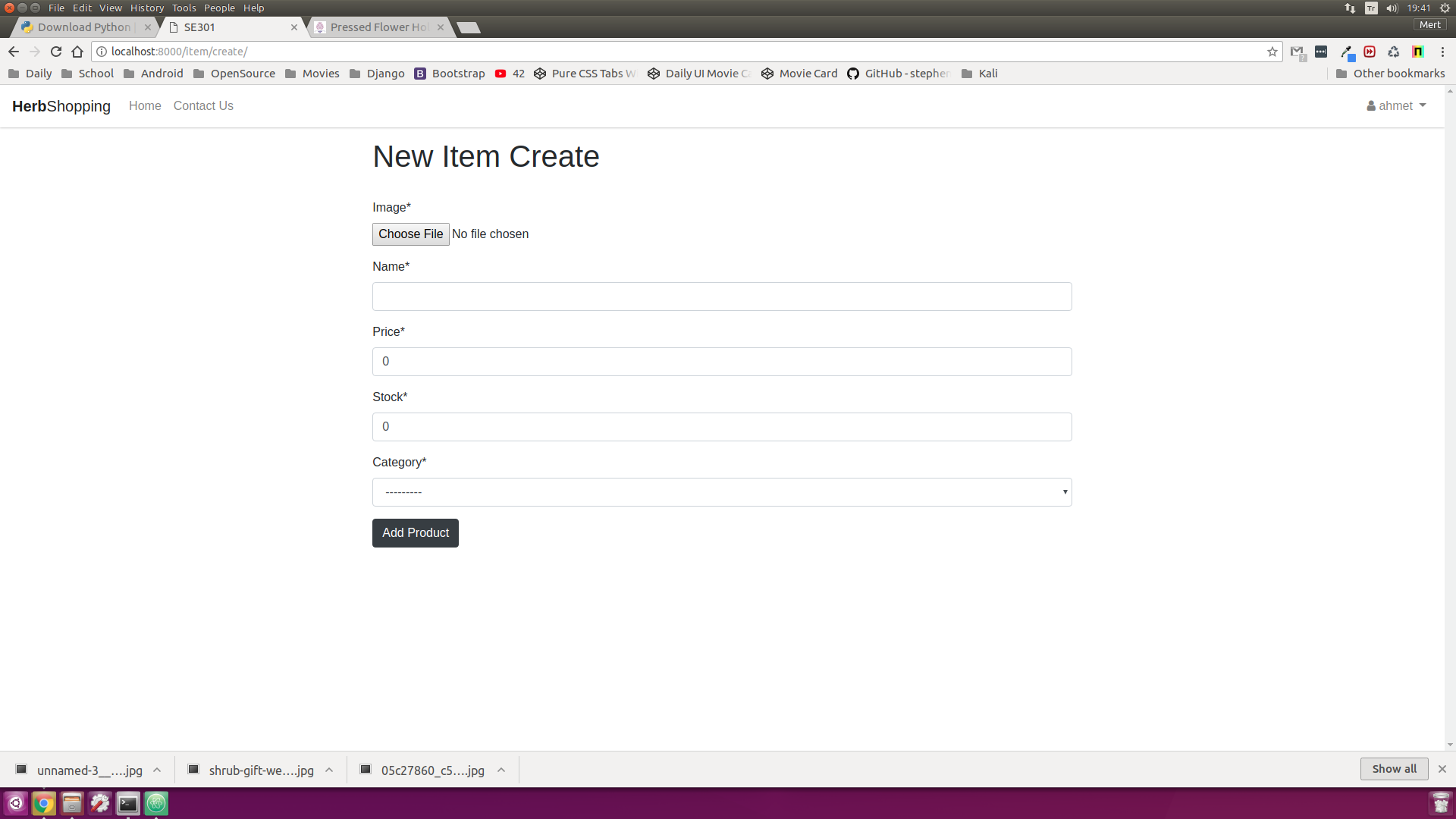
<https://developer.mozilla.org/en-US/.../Django/Introduction>

<https://github.com/django>

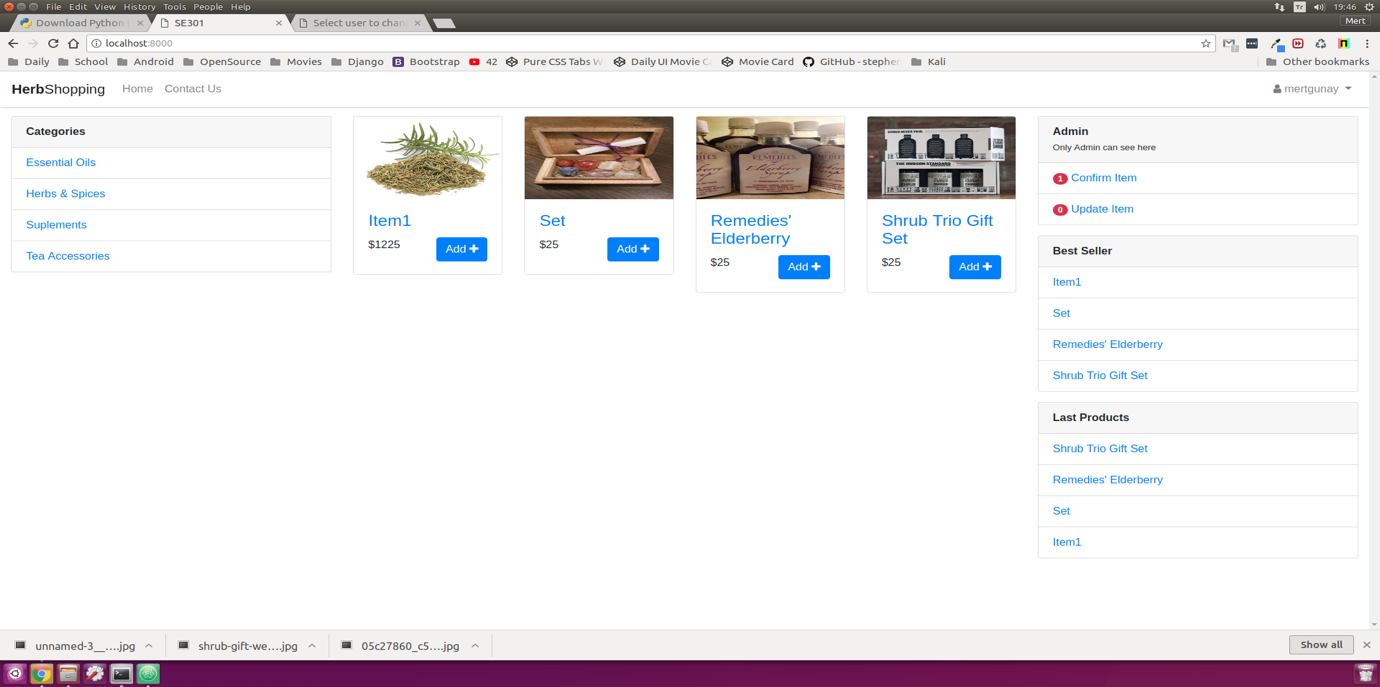
# Subsystem Services

We prefer to divide our system to subsystems to adapt each other themselves and, provide the whole system functionalities.

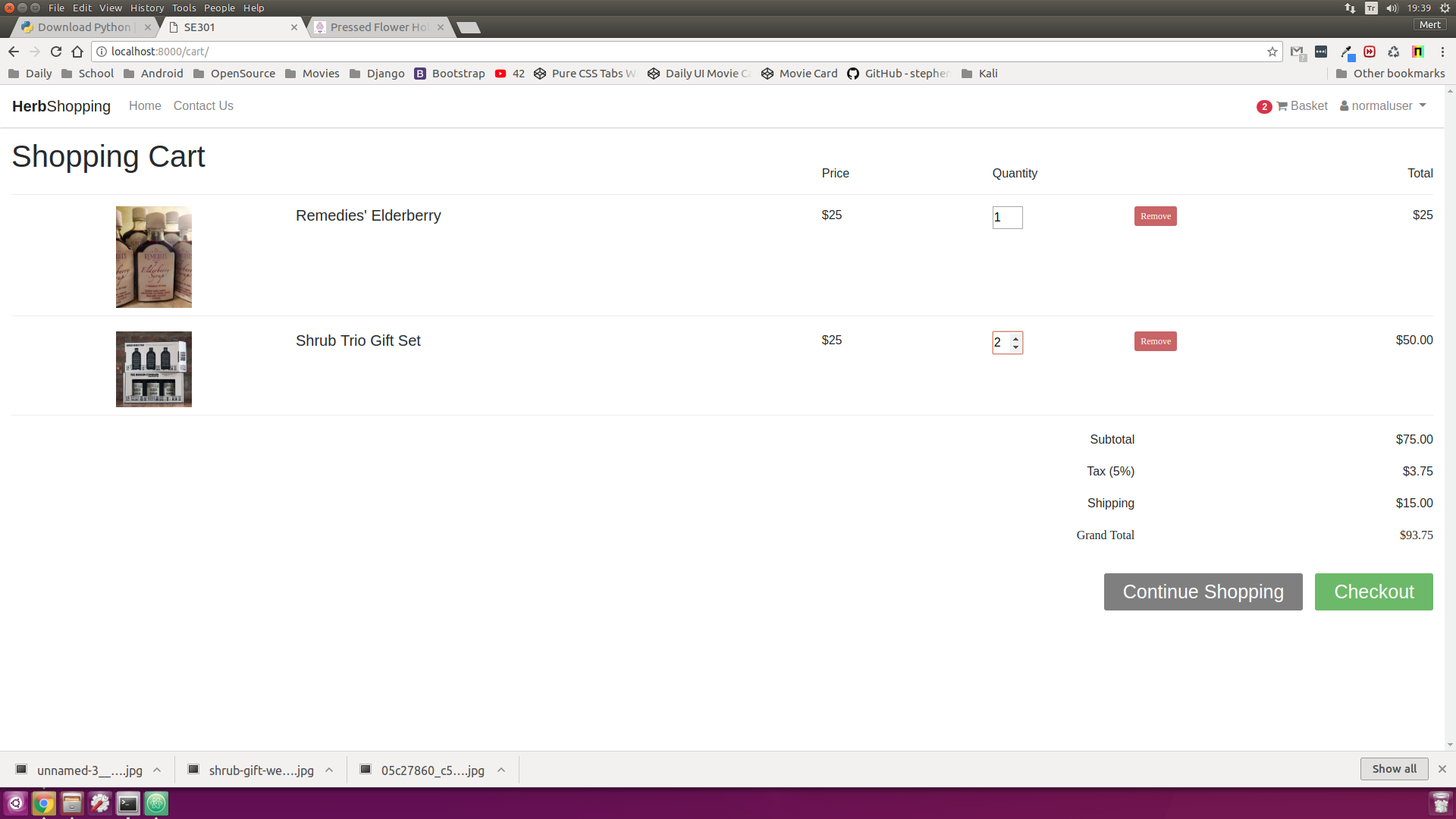
* **Add Item Interface;**



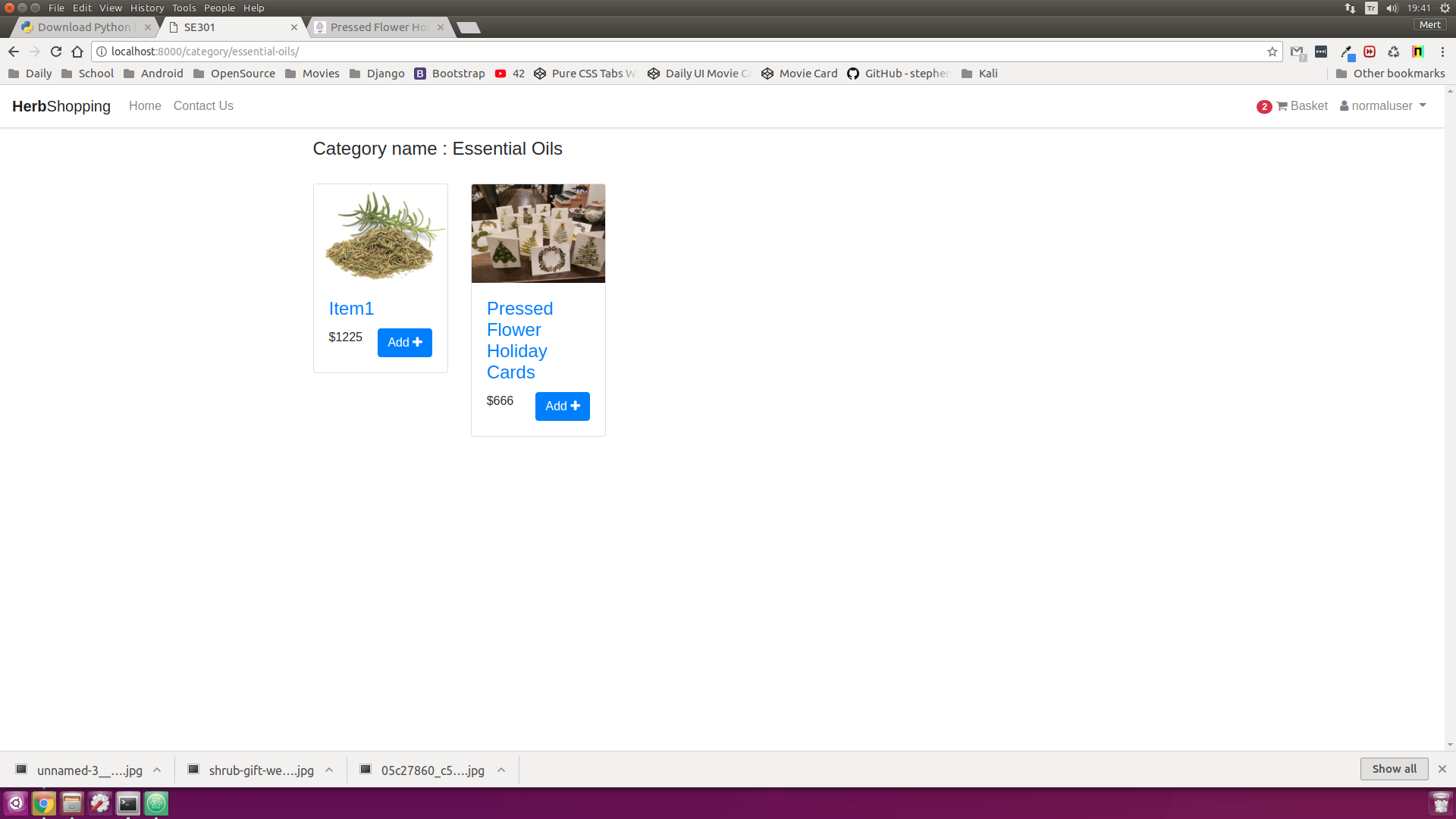
* **Admin Interface**



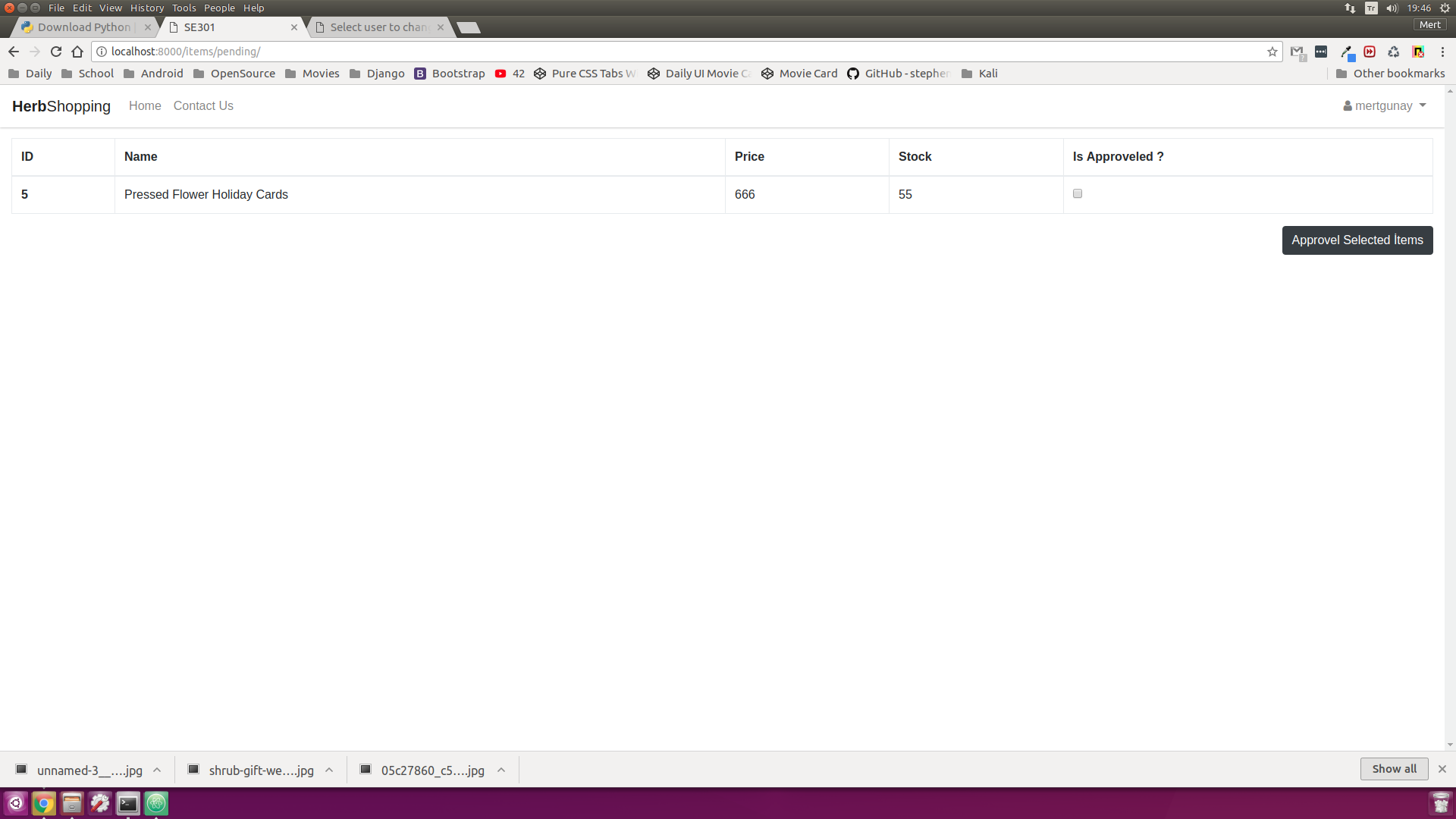
* **Basket Interface**



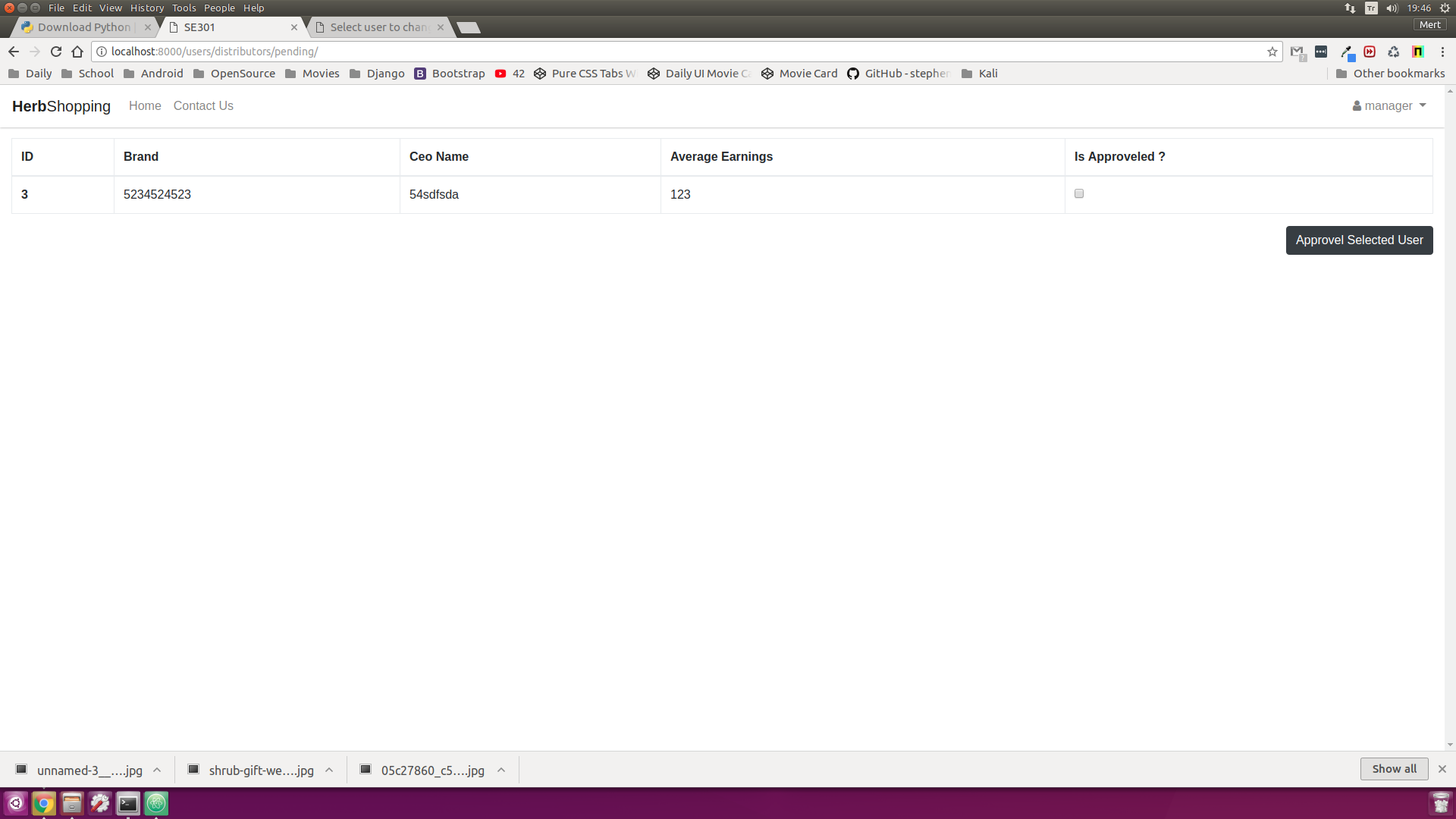
* **Category Interface**



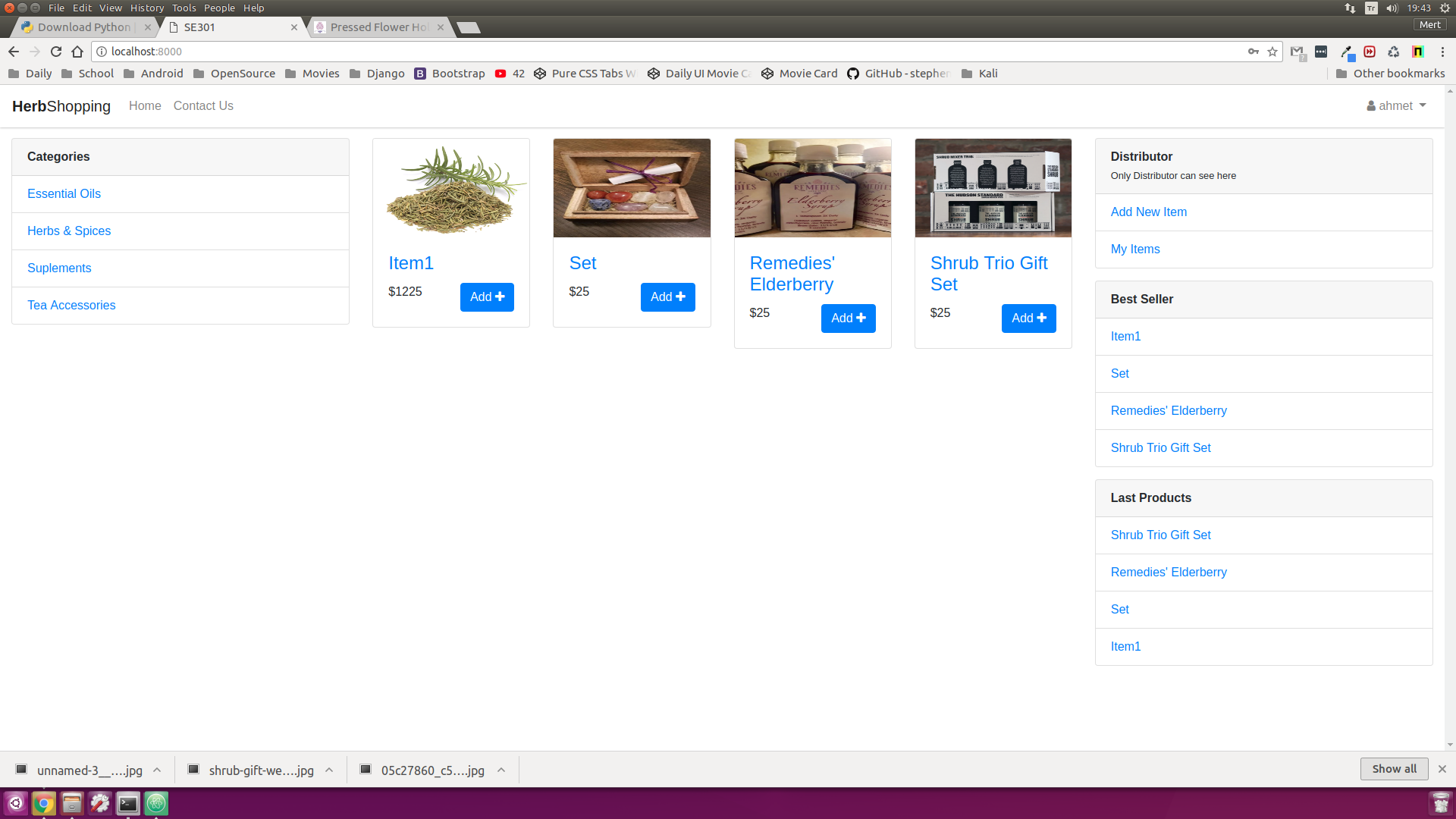
* **Approve Item Interface**



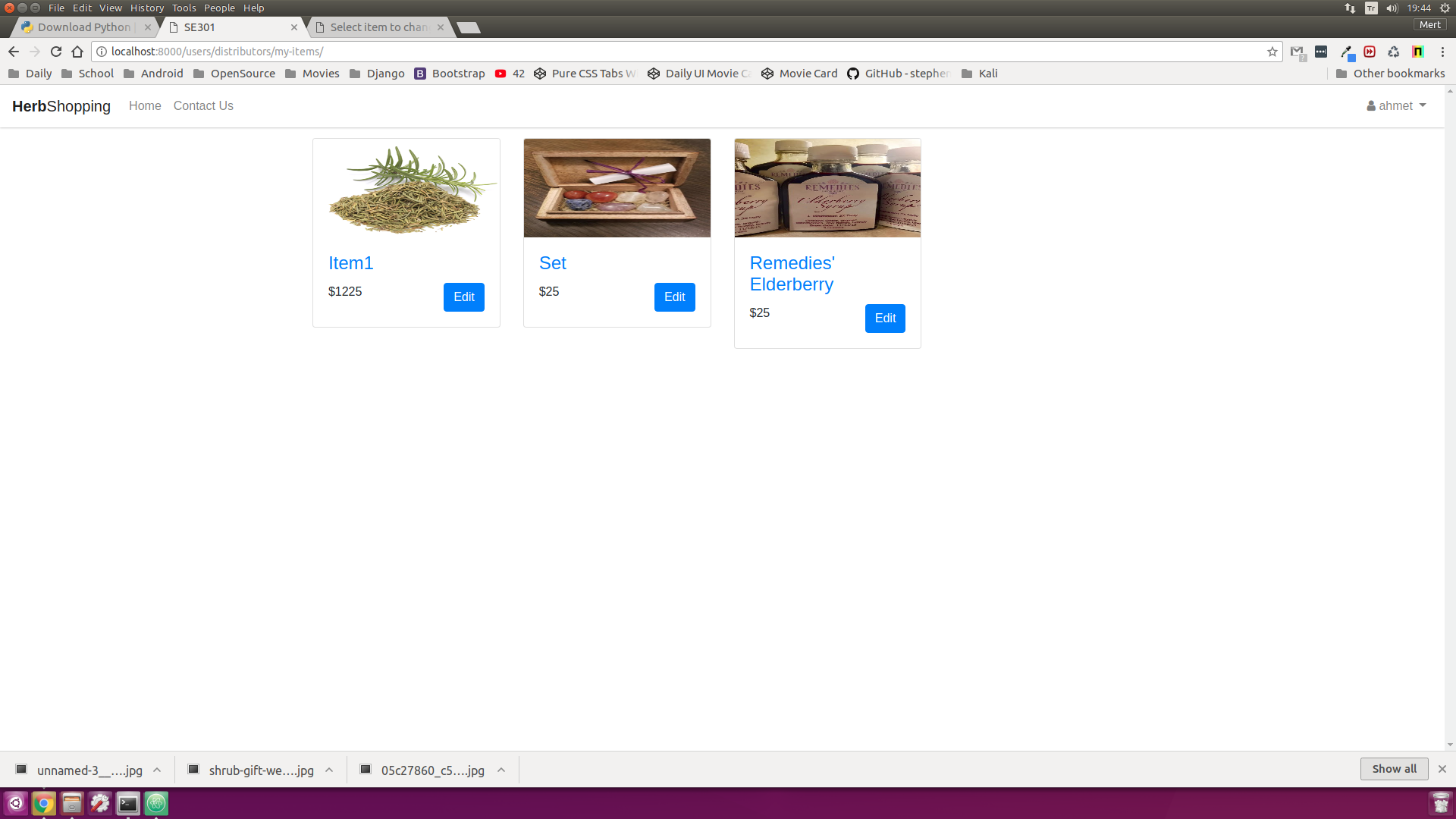
* **Distributor Accept Interface**



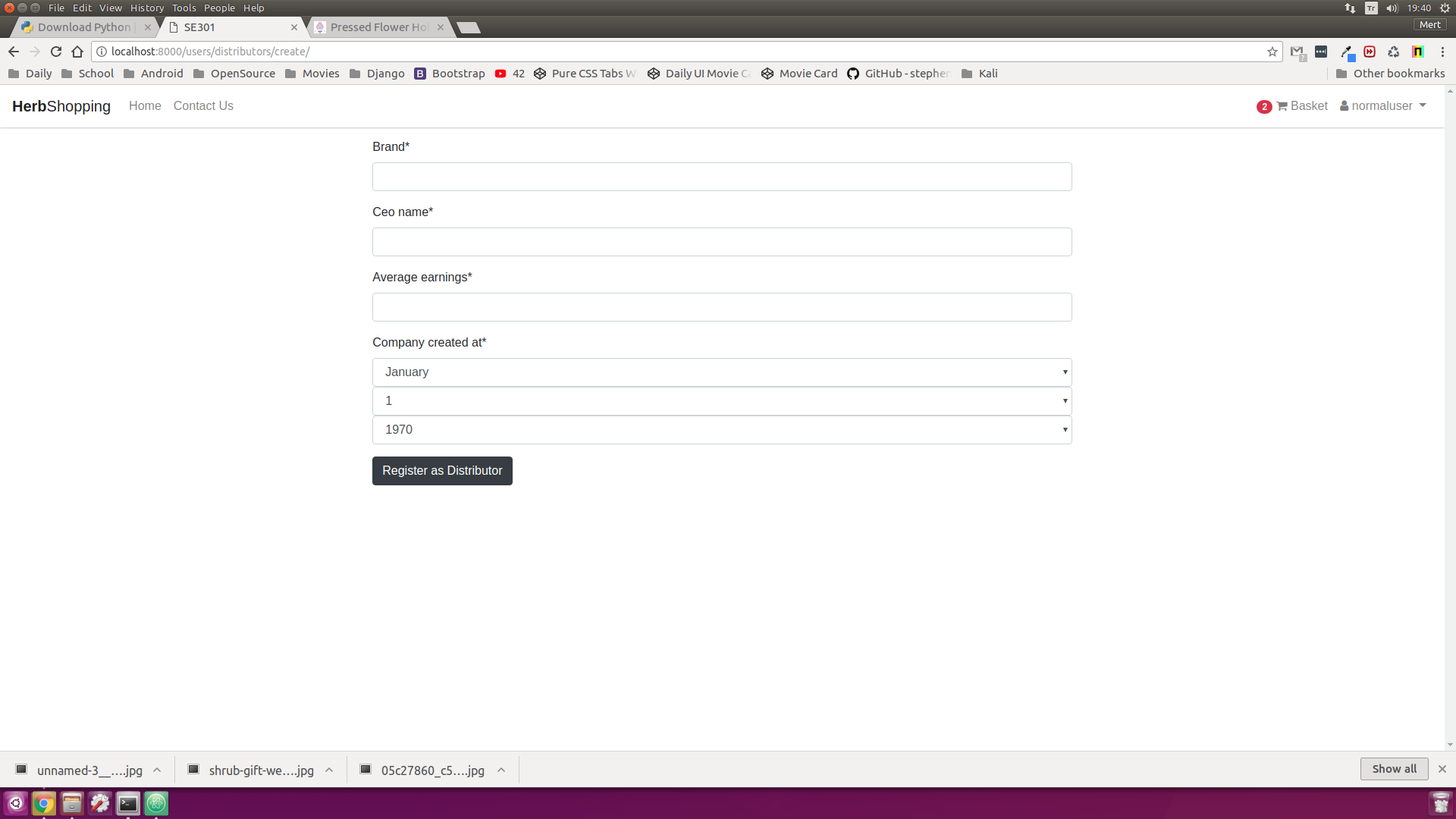
* **Distributor Interface**



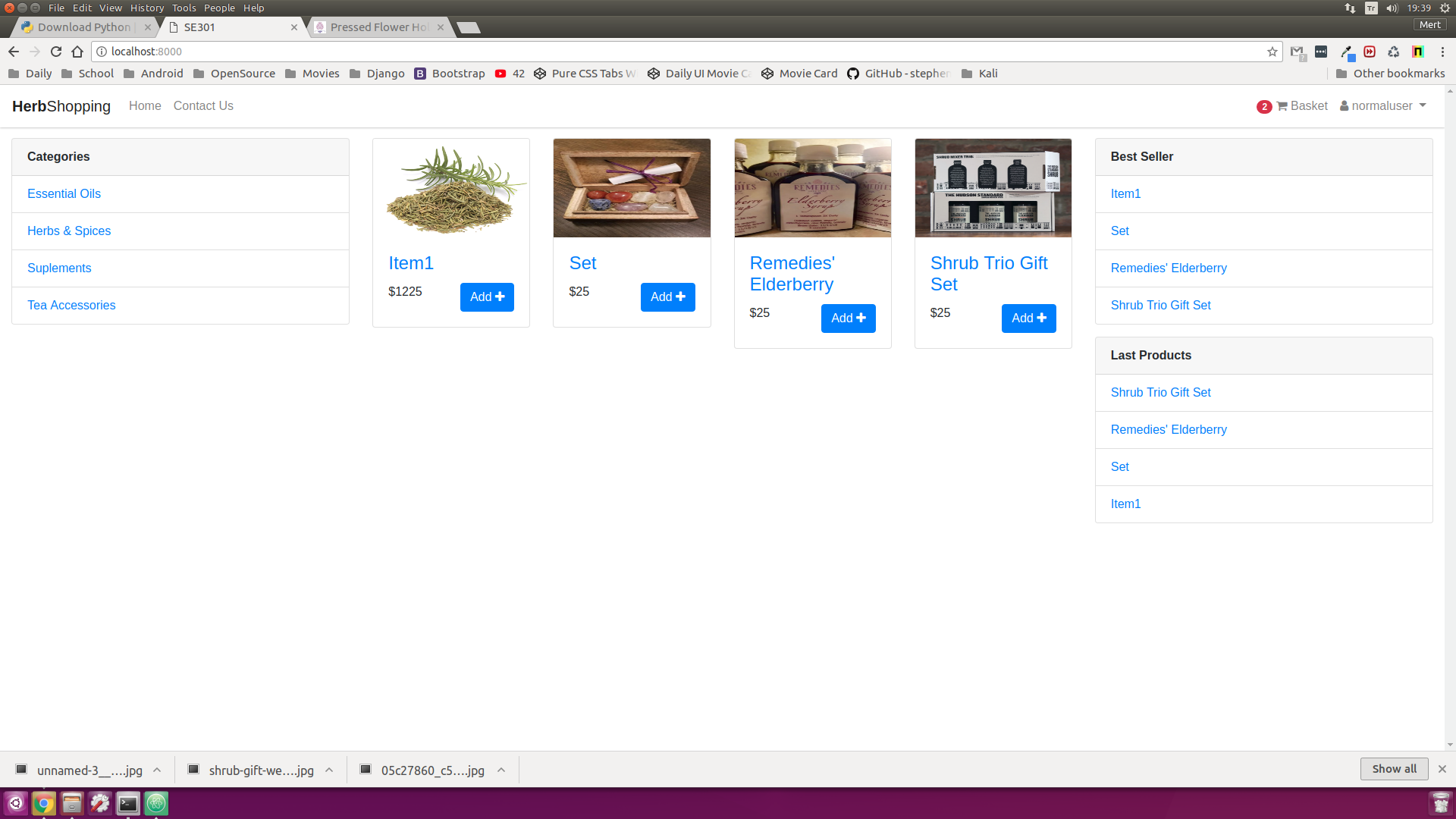
* **Items of Distributor Interface**



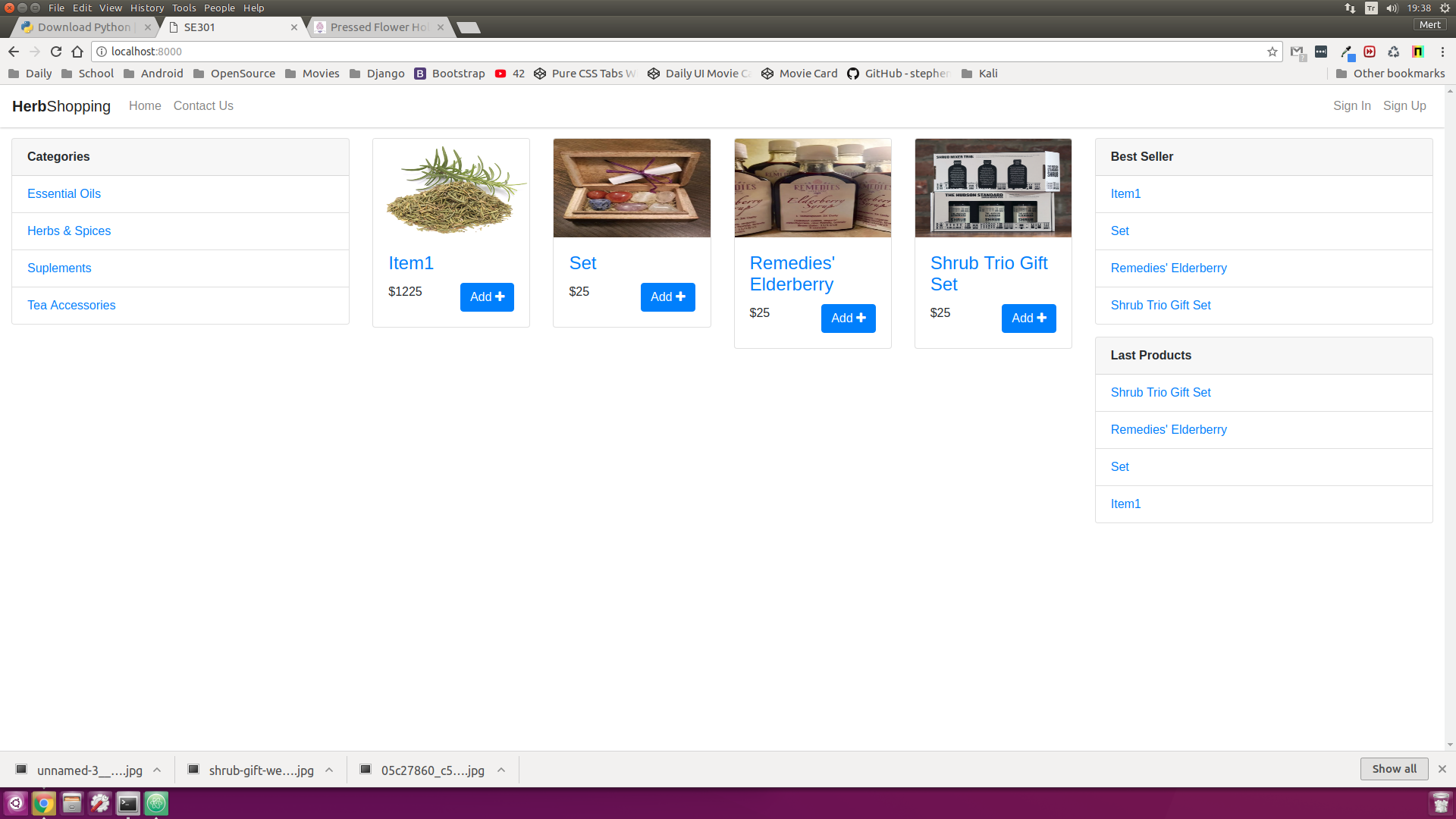
* **Distributor Register Interface**



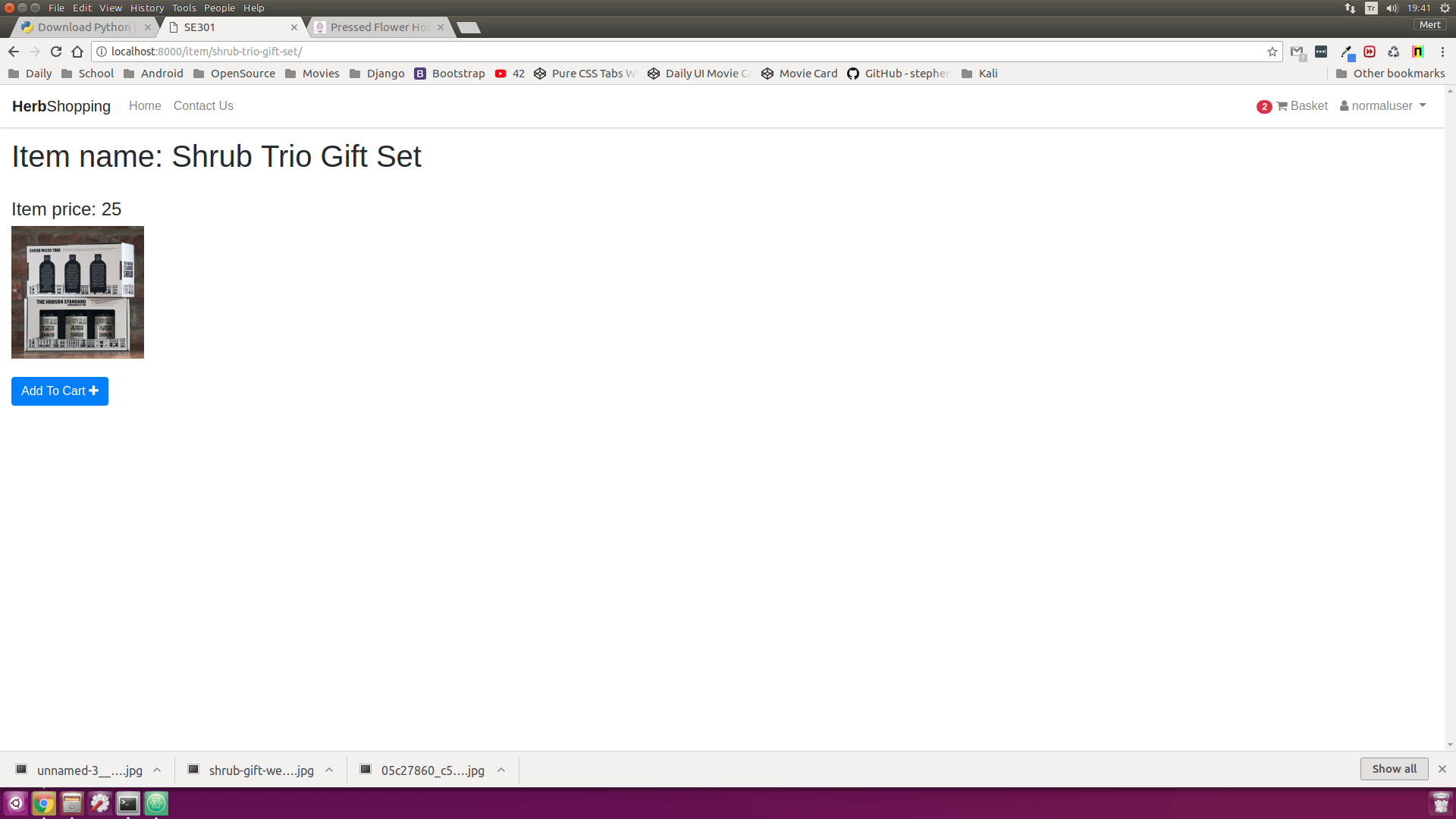
* **Web-site Main Page**



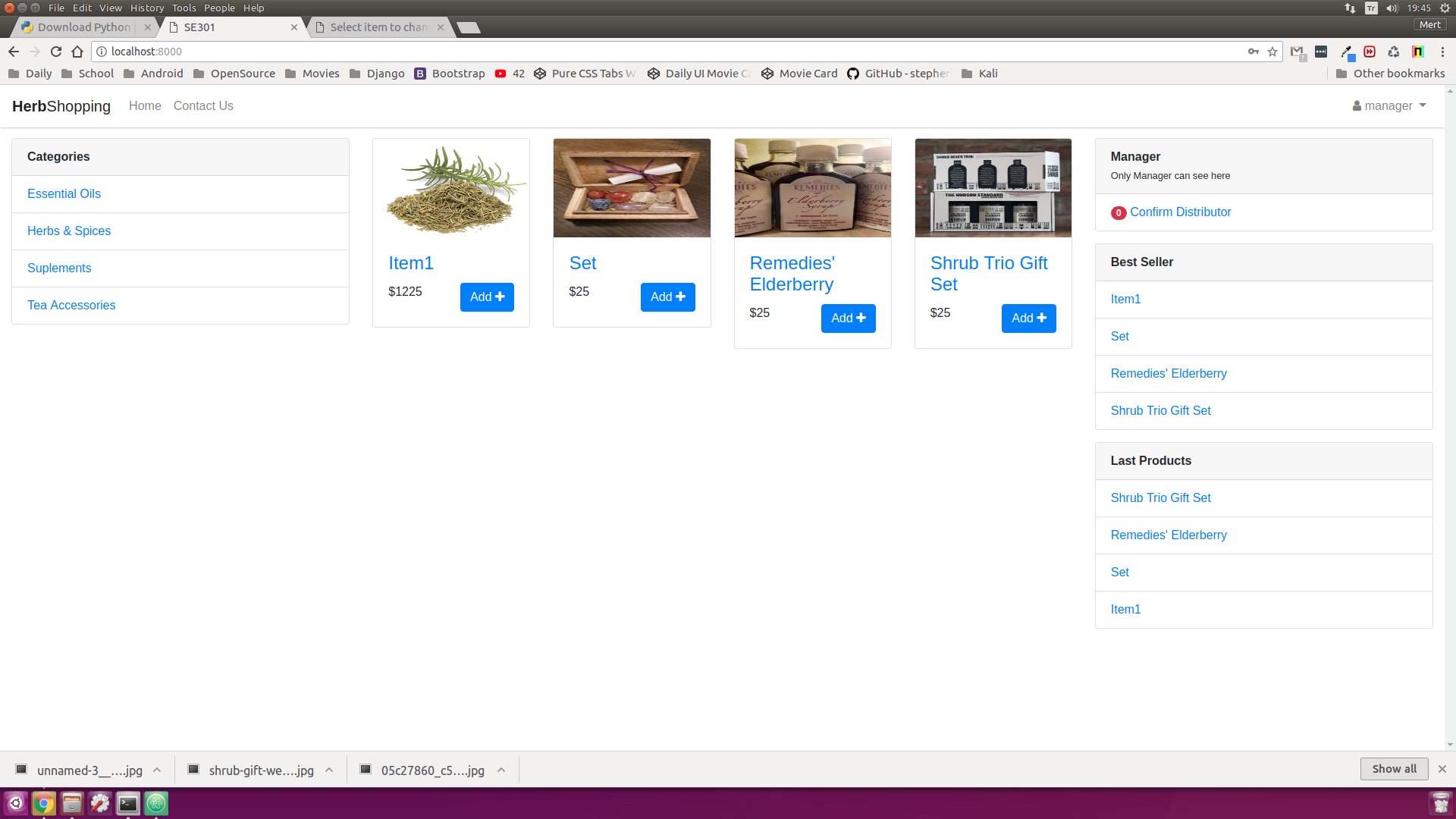
* **Main-page (Without Sign)**



* **Product Detail Interface**



* **Manager Home Page Interface**



# References

The following is an example of listing a book in this section. Check the text to see how it is cross referenced (The whole document is based on [1]).

1. Bruegge B. & Dutoit A.H.. (2010). *Object-Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall, 3rd ed.
2. <https://docs.djangoproject.com/en/2.0/ref/databases/>
3. https://www.uml-diagrams.org/composite-structure-diagrams.html