Deggendorf Institute of Technology

Faculty of Electrical Engineering, Media Technology and Computer Science

M.Sc. Electrical Engineering and Information Technology

Designing Inventory Project

(Lap Inventory)

A Final report for the subject:

Advanced Programming Techniques

Examiner

Prof. Dr. Andreas Fischer

Submitted by:

Muhammad Danyal Butt (772823)

Tejas Buva (0096201)

Mahmoud Gahelrasoul (775836)

Submitted date: 24/01/2020

Content

1 Introduction

This project is basically an inventory Web Application. The name of this web-app is “Lap Inventory”. Aim is to develop a software package that can be used to keep an electronic (online) inventory. This web-app is designed for the users of different branches to add or update their products in the inventory. In our Inventory system it will help to keep track of all the company assets. Multiple users are interacting with the inventory.

Administrators create and remove accounts for employees and external accountants. Employees belonging to a certain branch can enter items, specifying number, value, and category. External accountants can get an inventory overview. We have used python and Django framework for backend and for front end we used HTML, CSS and JavaScript.

2 Programming Language

In this project we used the Python as a programming language. The version of python is 3.7.2. In technical terms, Python is an object-oriented, high-level programming language with integrated dynamic semantics primarily for the web-app development. It is highly attractive in the field of Rapid Application Development because python offers dynamic typing and dynamic binding options.

Python is relatively simple, so it is easy to learn as it has a unique syntax that focuses on readability like natural language. Developers can read and translate Python code much easier than other languages. In turn, this reduces the cost of program maintenance and development because it allows teams to work collaboratively without significant language and experience barriers.

Additionally, Python supports the use of modules and packages, which means that programs can be designed in a modular style and code can be reused across a variety of projects. Once you've developed a module or package you need, it can be scaled for use in other projects, and it's easy to import or export these modules. One of the most promising benefits of Python is that both the standard library and the interpreter are available free of charge, in both binary and source form. There is no exclusivity either, as Python and all the necessary tools are available on all major platforms. Therefore, it is an enticing option for developers who don't want to worry about paying high development costs.

3 Database and Framework

In Lap Inventory, application design Django 2.2 is used. It is the latest version that was available at the time of design. Django is a free and open source web application framework, written in Python. A web framework is a set of components that helps you to develop websites faster and easier.

When you are building a website, you always need a similar set of components: a way to handle user authentications like a user signing up, signing in, and signing out, a management panel for your website, forms, a way to upload files, etc. To understand what Django is for, we need to take a closer look at the servers. The first thing is that the server needs to know that you want it to serve you a web page.

When a request comes to a web app server, it is passed to Django that tries to figure out what is requested. Django takes a web page address first and tries to figure out what to do. This part is done by Django's urlresolver (note that a website address is called a URL – Uniform Resource Locator – so the name urlresolver makes sense). It is not very smart, that is, it takes a list of patterns and tries to match the URL. Django checks patterns from top to bottom and if something is matched, then Django passes the request to the associated function.

In the view function, all the interesting things are done; we can look at a database for some information. Maybe the user requested to change something in the data? Like a letter saying, "Please change the description of my article." The view function can check whether you are allowed to do that or not, if you are then it updates the article description for you and sends back a message: "Done!” as a response generated from Django to can send back to the user's web browser.

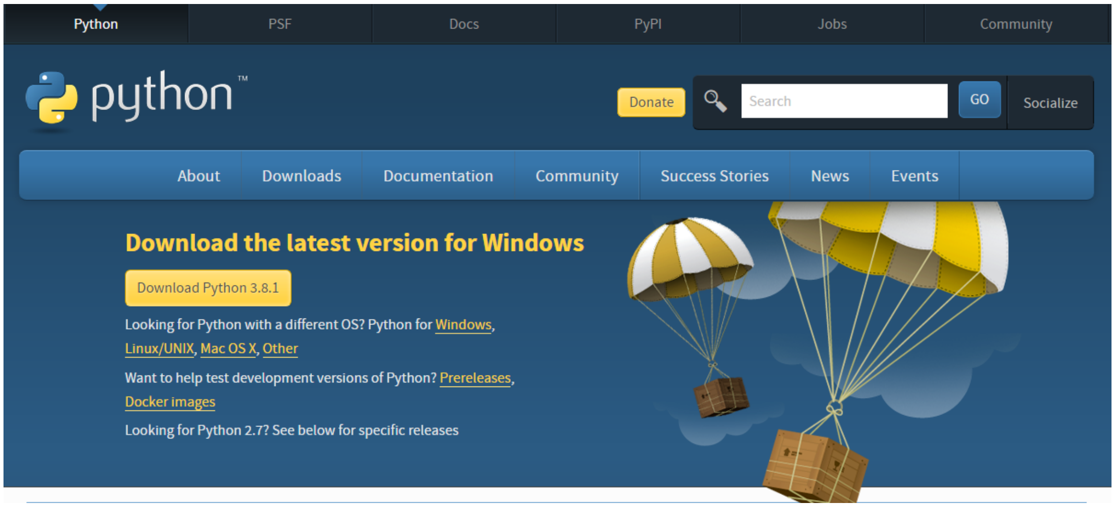
4 Installation instructions:

In this chapter we will illustrate how to install Django in your PC or MAC, firstly we should know that Django is a Python web framework, thus requiring Python to be installed on your machine. At this project we installed Python 3.8.

To install Python on your machine, go to https://python.org/downloads/. The website should offer you a download button for the latest Python version. Download the executable installer. Check Install launcher for all users (recommended) and Add Python 3.8 to Path then click Install Now.

4.1 Downloading:

The following page will appear in your browser.



Download the latest version for Windows

Looking for Python with a different OS? Python for Windows, Linux/UNIX, Mac OS X, Other

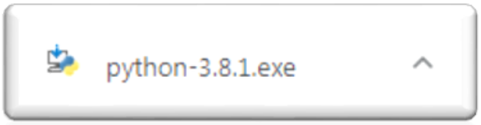
Want to help test development versions of Python? Prereleases, Docker images

Looking for Python 2.7? See below for specific releases

Click the Download Python 3.8.1 button

The file named python-3.8.1.exe should start downloading into your standard download folder. This file is about 30 Mb so it might take a while to download fully if you are on a slow internet connection (it took me about 10 seconds over a cable modem).

The file should appear as



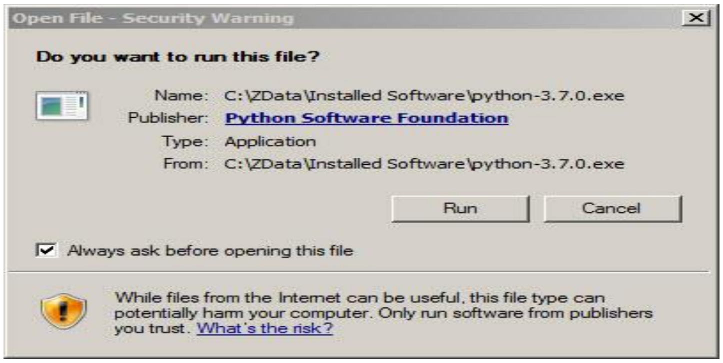
Move this file to a more permanent location, so that you can install Python (and reinstall it easily later, if necessary).

Feel free to explore this webpage further; if you want to just continue the installation, you can terminate the tab browsing this webpage.

Start the Installing instructions directly below.

4.2 Installing:

Double-click the icon labeling the file python-3.8.0.exe. An Open File - Security Warning pop-up window will appear.



Open File - Security Warning

Do you want to run this file?

Name: C:\ZData\Installed Software\python-3.7.0.exe
Publisher: Python Software Foundation
Type: Application
From: C:\ZData\Installed Software\python-3.7.0.exe

☑ Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust. What's the risk?

Security Warning Window

2.Click Run A Python 3.8.0 (32-bit) Setup pop-up window will appear.

3.If the Python Installer finds an earlier version of Python installed on your computer, the Install Now message may instead appear as Upgrade Now (and the checkboxes will not appear).

4.Highlight the Install Now (or Upgrade Now) message, and then click it.

5.A User Account Control pop-up window will appear, posing the question Do you want the allow the following program to make changes to this computer?

6.Click the Yes button.

7.A new Python 3.8.0 (32-bit) Setup pop-up window will appear with a Setup Progress message and a progress bar.

During installation, it will show the various components it is installing and move the progress bar towards completion. Soon, a new Python 3.8.0 Setup pop-up window will appear with a Setup was successfully message.

Click the Close button Python should now be installed.

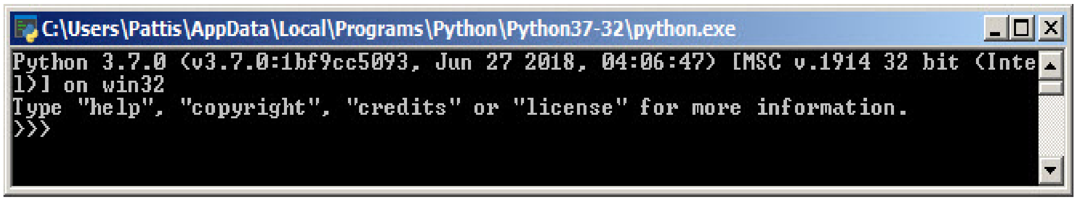
4.3 Verifying

To try to verify installation,

Navigate to the directory C:\Users\Pattis\AppData\Local\Programs\Python\Python38-32 (or to whatever directory Python was installed: see the pop-up window for Installing step 3).

Double-click the icon/file python.exe.

The following pop-up window will appear.



```
C:\Users\Pattis\AppData\Local\Programs\Python\Python37-32\python.exe
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:06:47) [MSC v.1914 32 bit (Inte
l)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

pop-up window

Information about Python version which can be used with Django:

Django version	Python versions
1.11	2.7, 3.4, 3.5, 3.6, 3.7,3.8 (added in 1.11.17)
2.0	3.4, 3.5, 3.6, 3.7
2.1, 2.2	3.5, 3.6, 3.7

Python 3 is recommended because Django 1.11 is the last version to support Python 2.7. Support for Python 2.7 and Django 1.11 ends in 2020.

Since newer versions of Python are often faster, have more features, and are better supported, the latest version of Python 3 is recommended.

You don’t lose anything in Django by using an older release, but you don’t take advantage of the improvements and optimizations in newer Python releases. Third-party applications for use with Django are, of course, free to set their own version requirements.

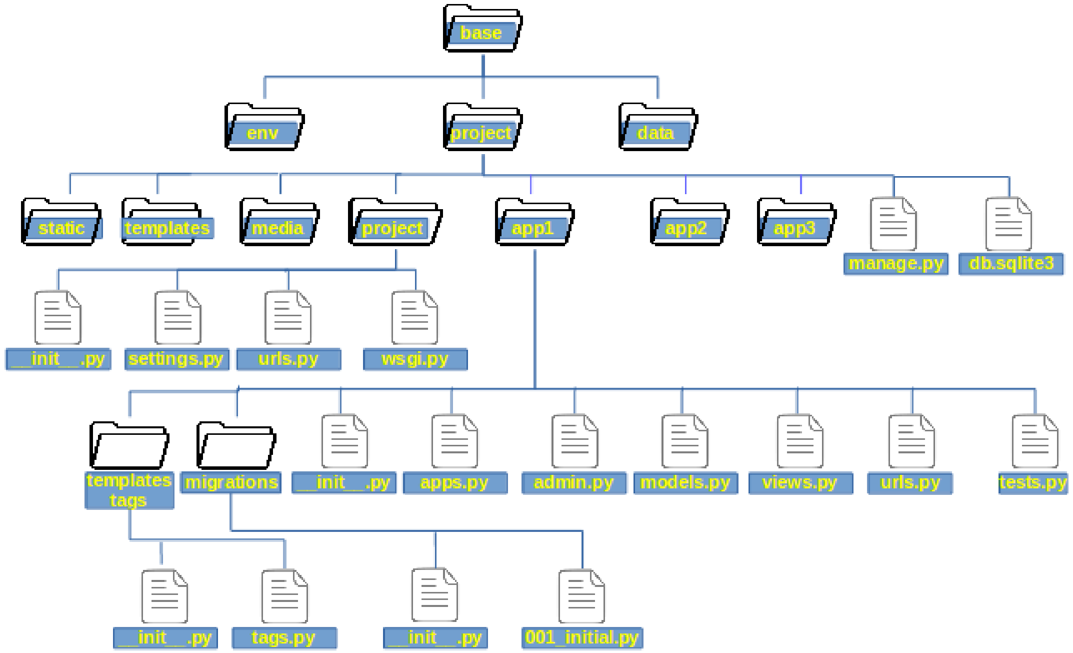
Following we should go for installing Django Framework.

4.4 Installing Django

4.4.1 Django’s prerequisites

Django requires Python. See the next diagram to understand how Django work. in the next question for the versions of Python that work with each version of Django. Other Python libraries may be required for some uses, but you’ll receive an error about it as they’re needed.

For a development environment – if you just want to experiment with Django – you don’t need to have a separate Web server installed; Django comes with its own lightweight development server. For a production environment, Django follows the WSGI spec, PEP 3333, which means it can run on a variety of server platforms. See Deploying Django for some popular alternatives.



Django project layout and settings

If you want to use Django with a database, which is probably the case, you’ll also need a database engine. PostgreSQL is recommended, because we’re PostgreSQL fans, and MySQL, SQLite, and Oracle are also supported.

4.4.2 Stable version or development version

Generally, if you’re using code in production, you should be using a stable release. The Django project publishes a full stable release every nine months or so, with bugfix updates in between. These stable releases contain the API that is covered by our backwards compatibility guarantees; if you write code against stable releases, you shouldn’t have any problems upgrading when the next official version is released.

4.4.3 About pip

Pip is a package manage for python. It makes installing and uninstalling python packages (such as Django!) very easy. For the rest of the installation, we’ll use pip to install python packages from the command line.

But keep in mind that pip is already installed if you are using python 2>=2.7 or python 3 >= 3.4 download from python.org or if you are working in a virtual environment created by virtualenv or pyvenv. Just make sure to upgrade pip.

4.4.4 Install virtualenv and virtualenvwrapper

virtualenv and virtualenvwrapper provide a dedicated environment for each Django project you create. While not mandatory, this is considered a best practice and will save you time in the future when you’re ready to deploy your project. Simply type: “pip install virtualenvwrapper-win”

Then create a virtual environment for your project: “mkvirtualenv myproject”

The virtual environment will be activated automatically, and you’ll see “(myproject)” next to the command prompt to designate that. If you start a new command prompt, you’ll need to activate the environment again using: “work on myproject”

4.4.5 Steps of installation:

After installing python, open the command prompt and check that the Python version matches the version you installed by executing: “python –version".

4.4.6 Installing an official release with pip:

This is the recommended way to install Django.

1. Install pip. The easiest is to use the standalone pip installer. If your distribu tion already has pip installed, you might need to update it if it’s outdated. If it’s outdated, you’ll know because installation won’t work.
2. Take a look at virtualenv and virtualenvwrapper. These tools provide isolat

ed Python environments, which are more practical than installing packages systemwide. They also allow installing packages without administrator privileges. The contributing tutorial walks through how to create a virtualenv.

3. After you've created and activated a virtual environment, enter the command: " ...\> pip install Django"

4.4.7 Verifying

After installation of Python and Django you should make sure that Django can be seen by Python, type python from your shell. Then at the Python prompt, try to import Django:

“>>> import django

>>> print(django.get_version()) 2.2"

Because You may have another version of Django installed.

4.4.8 Common mistake

If Django-admin only displays the help text no matter what arguments it is given, there is probably a problem with the file association in Windows. Check if there is more than one environment variable set for running Python scripts in PATH. This usually occurs when there is more than one Python version installed. We finished installing Python and Django and now we move for installing Libraries.

4.5 Installing Libraries:

4.5.1 Django-crispy-forms:

Django-crispy-forms provides you with a |crispy filter and {% crispy %} tag that will let you control the rendering behavior of your Django forms in a very elegant and DRY way. Have full control without writing custom form templates. All this without breaking the standard way of doing things in Django, so it plays nice with any other form application.

For Install latest stable version into your python path using pip or easy_install:

“pip install --upgrade django-crispy-forms”

Add crispy_forms to your INSTALLED_APPS in settings.py:

```
“INSTALLED_APPS = (

    ...
```

'crispy_forms',

)”

In production environments, always activate Django template cache loader. This is available since Django 1.2 and what it does is basically load templates once, then cache the result for every subsequent render. This leads to a significant performance improvement.

4.5.2 Django markdown:

is Django application that allows use markdown wysiwyg in flatpages, admin forms and other forms.Django markdown should be installed using pip:

“pip install django-markdown”

Add ‘django_markdown’ to INSTALLED_APPS

“INSTALLED_APPS += ('django_markdown',)”

Add django_markdown urls to base urls

“url('^markdown/', include('django_markdown.urls')),”

4.5.3 Pillow

Pillow is the friendly PIL fork by Alex Clark and Contributors. PIL is the Python Imaging Library by Fredrik Lundh and Contributors. It is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.

Pillow offers several standard procedures for image manipulation. These include:

- per-pixel manipulations.
- masking and transparency handling.
- image filtering, such as blurring, contouring, smoothing, or edge finding.
- image enhancing, such as sharpening, adjusting brightness, contrast or color.
- adding text to images and much more.

For installation use command: pip install Pillow

4.5.4 Django Tagging

A generic tagging application for Django projects, which allows association of several tags with any Django model instance and makes retrieval of tags simple.

For installation use command: pip install Django-tagging

4.5.5 Django-filter

Django-filter is a generic, reusable application to alleviate writing some of the more mundane bits of view code. Specifically, it allows users to filter down a query set based on a model’s fields, displaying the form to let them do this.

For installation use command: Pip install Django-filters

4.5.6 Django mixer

By default, Mixer try to generate fake data. If you want randomize values initialize the Mixer by manual like: Mixer(fake=False).

By default, Mixer saves generated objects in database. If you want to disable this, initialize the Mixer by manual like: Mixer(commit=False)

For installation use command: Pip install mixer

4.5.7 Bootstrap 4

Requirements Django >= 1.1

Steps of installation

1. Install using pip: “pip install django-bootstrap4”

Alternatively, you can install download or clone this repo and call

“pip install -e ..”

2. Add to INSTALLED_APPS in your settings.py: 'bootstrap4'

3. In your templates, load the bootstrap4 library and use the bootstrap_* tags:

Example template

```
“{% load bootstrap4 %}
 {# Display a form #}
 <form action="/url/to/submit/" method="post" class="form">
```

```
    {% csrf_token %}
    {% bootstrap_form form %}
    {% buttons %}
        <button type="submit" class="btn btn-primary">Submit</button>
    {% endbuttons %}
 </form>”
```

Now you are all set and ready to run the project.

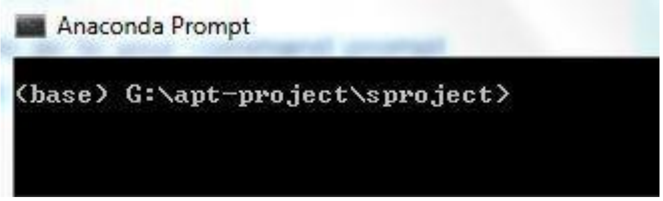
4.6 How it work

In this point we will illustrate how normal user can use the web based on simple way and in then I will explain the whole project through flowchart.

In order to run this application, you would need to install Python 3 and Django 2 on your machine. Apart from that you would also need a Command Prompt like Anaconda Prompt or any other command line interface of your choice.

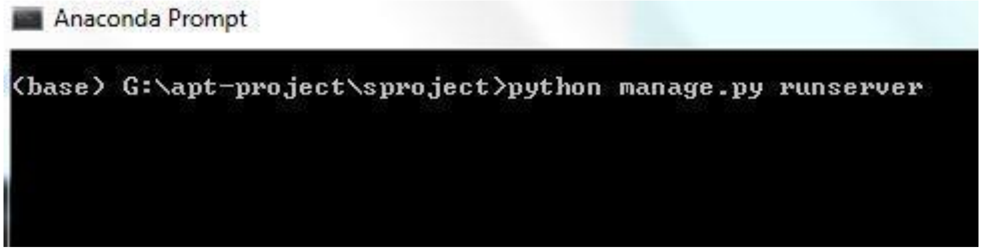
If you have all these things ready, you would need to go to your command prompt and give the location of the folder where your project is located.

For example: My project is in my machine as: G:\apt-project\sproject



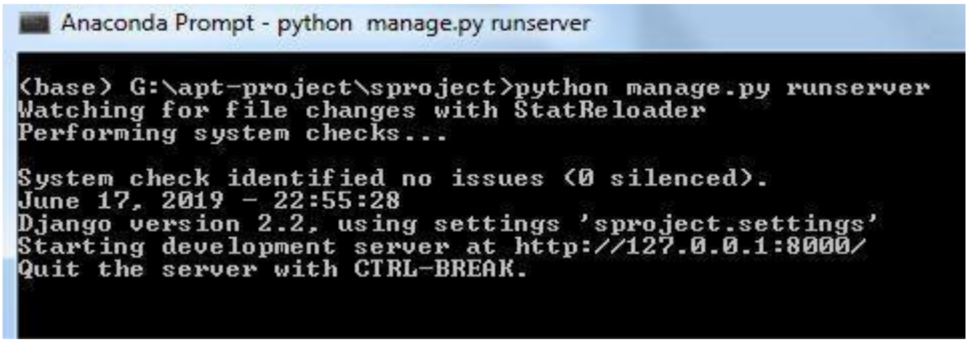
```
(base) G:\apt-project\sproject>
```

Now in order to run this project you need to run a command that is mentioned bellow.



```
(base) G:\apt-project\sproject>python manage.py runserver
```

When this command is executed it would give you a local address where you can view the layout of this application.



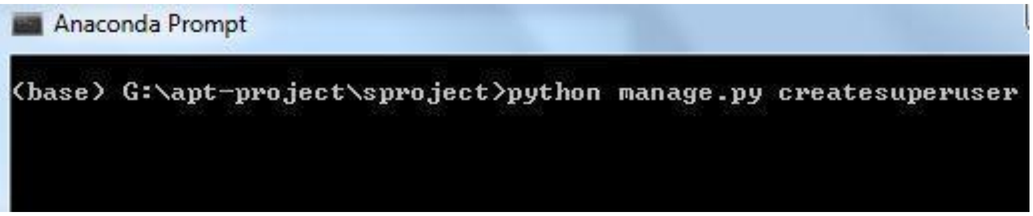
Anaconda Prompt - python manage.py runserver

```
(base) G:\apt-project\sproject>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 17, 2019 - 22:55:28
Django version 2.2, using settings 'sproject.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

By putting this address http://127.0.0.1:8000 you can enter the website.

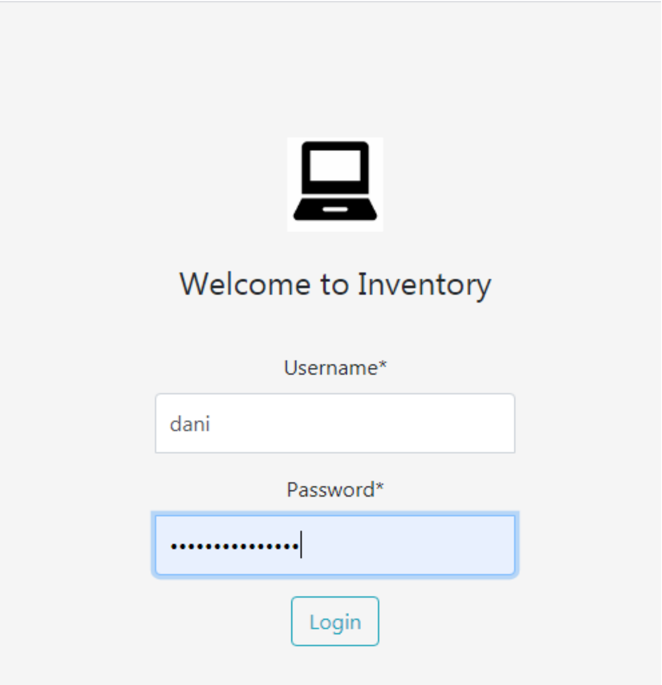
If you are the admin of the Website and you want to make Username and Password for the Admin. You would need to run the following command.



Anaconda Prompt

```
(base) G:\apt-project\sproject>python manage.py createsuperuser
```

After executing this command, it would ask you a suitable Username and Password and it would set it for the Admin login in the website.



Welcome to Inventory

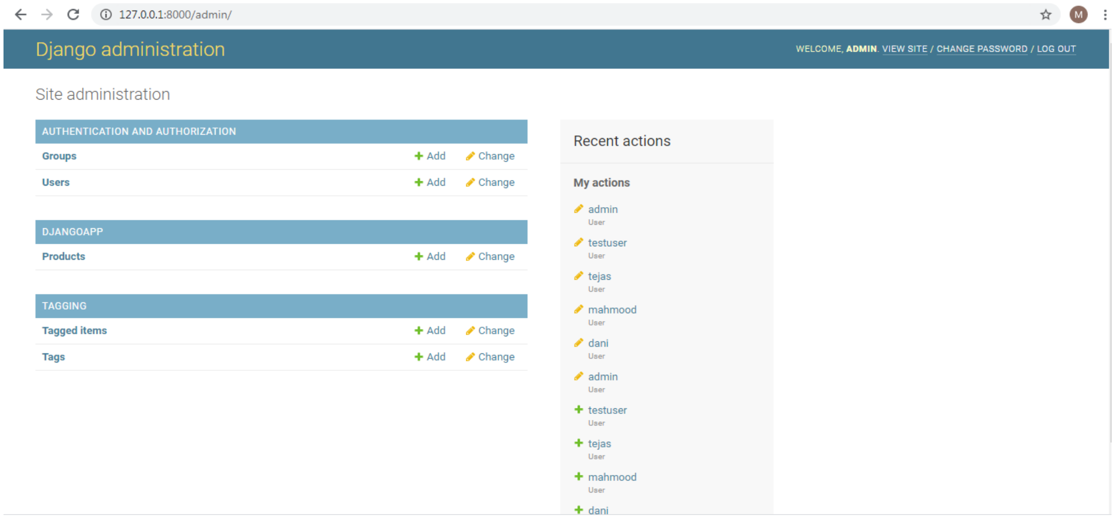
Username*

dani

Password*

•••••••••••••••

Login page which can (Admin , Employee and External accountant) login by username and password



127.0.0.1:8000/admin/

Django administration

Site administration

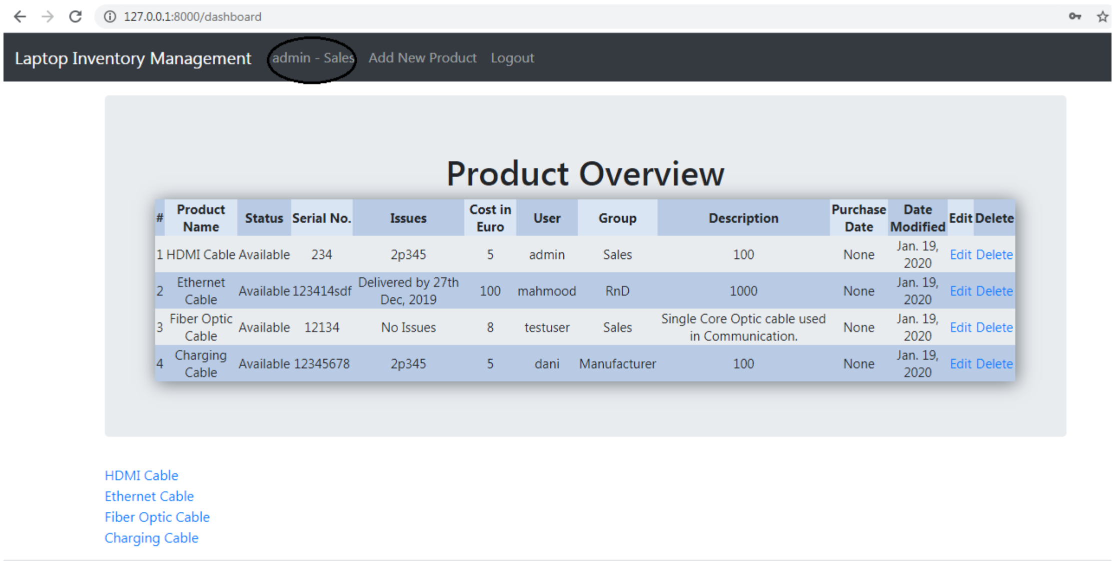
AUTHENTICATION AND AUTHORIZATION		
Groups	+ Add	Change
Users	+ Add	Change
DJANGOAPP		
Products	+ Add	Change
TAGGING		
Tagged items	+ Add	Change
Tags	+ Add	Change

Recent actions

My actions

- admin User
- testuser User
- tejas User
- mahmood User
- dani User
- admin User
- + testuser User
- + tejas User
- + mahmood User
- + dani

The admin can be logged into the application either by clicking the Admin login button or putting this address http://127.0.0.1:8000/admin/ Here the built-in Django administration functionality has been used. The admin will be the super user of the application, admin can anytime change the role of a user, delete any product or group anytime, can also delete any tags, also admin has the authority to cancel the registration of any user at any time.

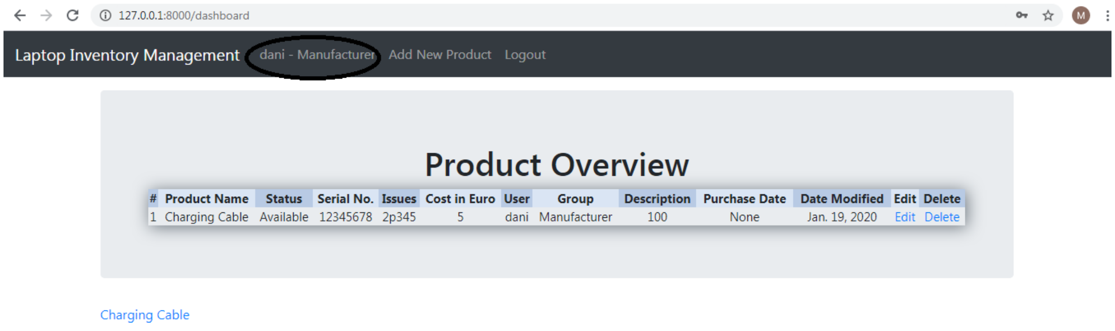


Product Overview

#	Product Name	Status	Serial No.	Issues	Cost in Euro	User	Group	Description	Purchase Date	Date Modified	Edit	Delete
1	HDMI Cable	Available	234	2p345	5	admin	Sales	100	None	Jan. 19, 2020	Edit	Delete
2	Ethernet Cable	Available	123414sdf	Delivered by 27th Dec, 2019	100	mahmood	RnD	1000	None	Jan. 19, 2020	Edit	Delete
3	Fiber Optic Cable	Available	12134	No Issues	8	testuser	Sales	Single Core Optic cable used in Communication.	None	Jan. 19, 2020	Edit	Delete
4	Charging Cable	Available	12345678	2p345	5	dani	Manufacturer	100	None	Jan. 19, 2020	Edit	Delete

HDMI Cable
Ethernet Cable
Fiber Optic Cable
Charging Cable

Admin interface show all product in all group's

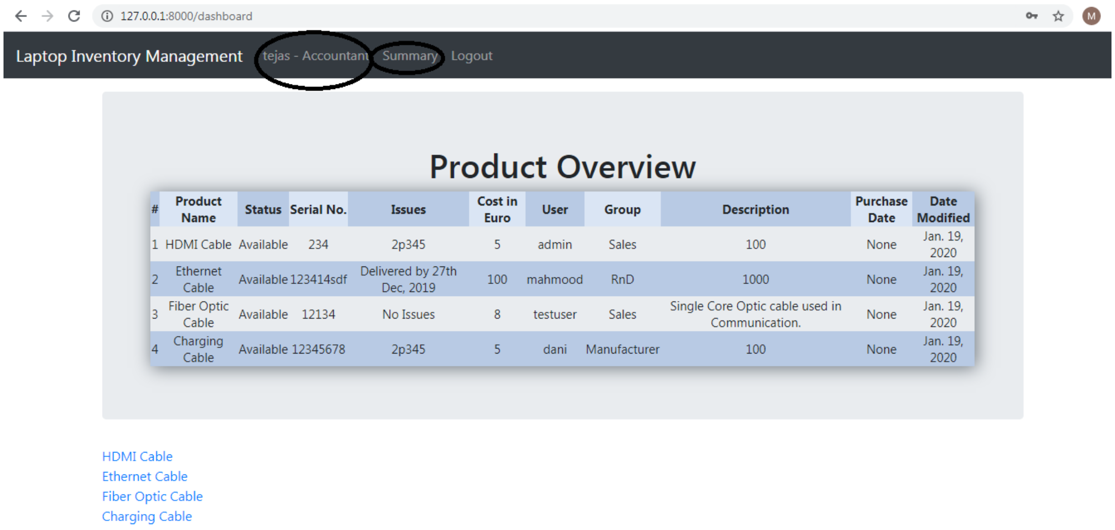


Product Overview

#	Product Name	Status	Serial No.	Issues	Cost in Euro	User	Group	Description	Purchase Date	Date Modified	Edit	Delete
1	Charging Cable	Available	12345678	2p345	5	dani	Manufacturer	100	None	Jan. 19, 2020	Edit	Delete

Charging Cable

Employee interface show all product in his group , the name of employee and his group in navigation bar and he can Edit or delete from right side anchor element

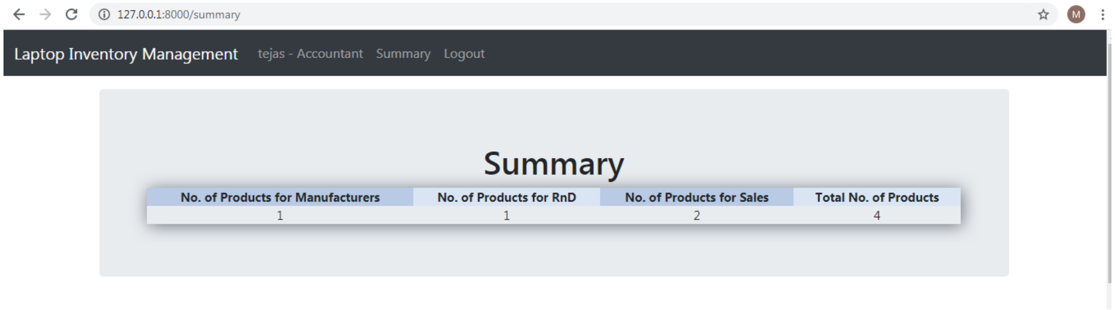


Product Overview

#	Product Name	Status	Serial No.	Issues	Cost in Euro	User	Group	Description	Purchase Date	Date Modified
1	HDMI Cable	Available	234	2p345	5	admin	Sales	100	None	Jan. 19, 2020
2	Ethernet Cable	Available	123414sdf	Delivered by 27th Dec, 2019	100	mahmood	RnD	1000	None	Jan. 19, 2020
3	Fiber Optic Cable	Available	12134	No Issues	8	testuser	Sales	Single Core Optic cable used in Communication.	None	Jan. 19, 2020
4	Charging Cable	Available	12345678	2p345	5	dani	Manufacturer	100	None	Jan. 19, 2020

HDMI Cable
Ethernet Cable
Fiber Optic Cable
Charging Cable

External accountant interface can show all product in all group's without any authority to add or remove but he can access to summary page



127.0.0.1:8000/summary

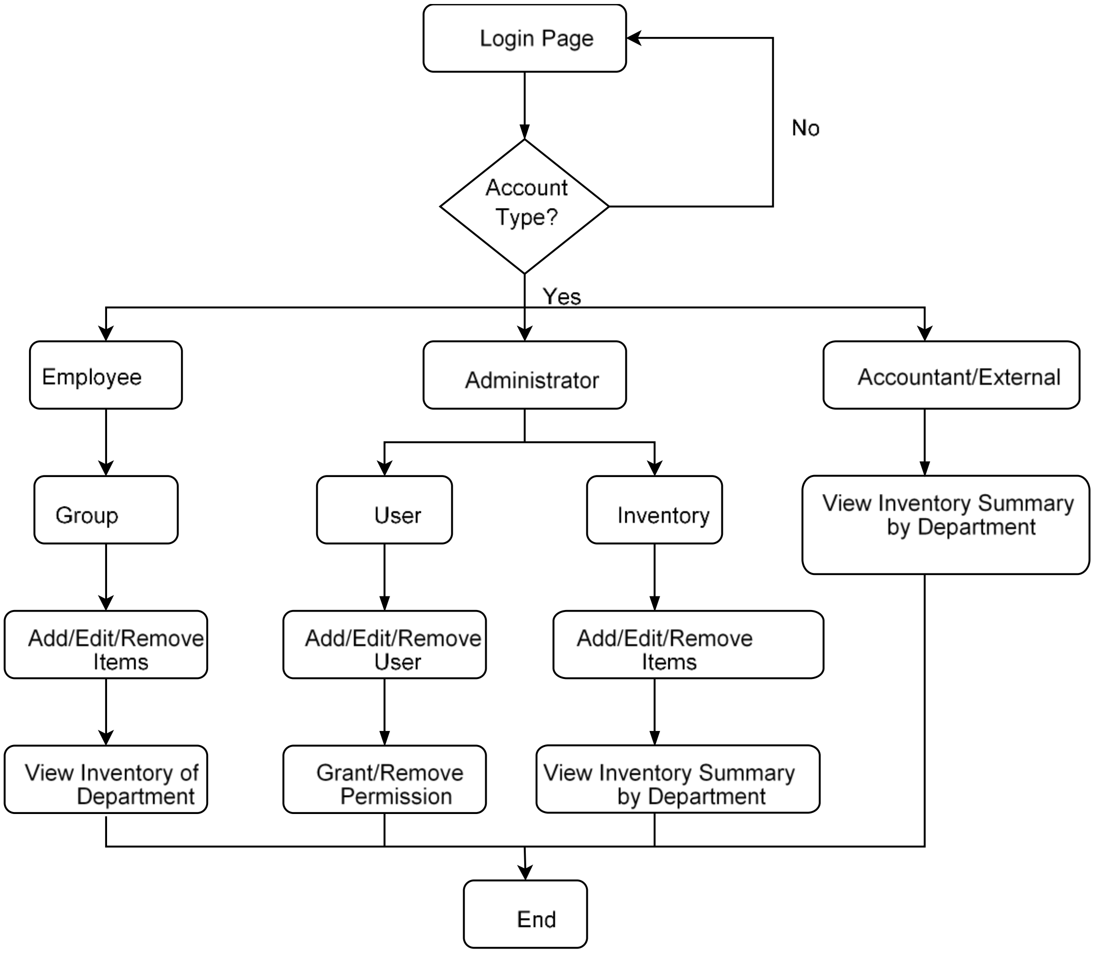
Laptop Inventory Management tejas - Accountant Summary Logout

Summary

No. of Products for Manufacturers	No. of Products for RnD	No. of Products for Sales	Total No. of Products
1	1	2	4

External accountant interface can see product summary

The next flowchart will explain how inventory work :



CHAPTER 5 TESTING

This chapter includes the methods that were used for testing, validating, and evaluating the system.

5.1 Methodology:

With this testing approach, all the specs were ready for a prototype, and a plan was already built to be shown; the tester started writing his or her code and saw if he or she could obtain the same results that the specs mentioned. This way, the specs were tested on each prototype, and continuous testing was applied. This also helped to minimize the testing that would have to be implemented at the end of the software lifecycle. In the process, all aspects of the software were tested. Steps to follow while implementing the methodology are as follows:

1. Start with a base function that you want to implement.

2. Create a document with the detailed requirement definition, an activity diagram with a description of the flow, database tables to be used, a component diagram, and a description of each component with the precondition and tables that would be affected by the component.

3. Give the document to the tester, and work with the tester while he or she writes the code to check if the steps in the document can be implemented and if the result of each use case can be achieved.

4. If the tester finds a step difficult to implement or thinks he or she is missing additional information to implement the functionality, then go to step 2; otherwise, go to step 3.

5. Ask the tester to log on all the errors and difficulties he or she encountered while working on the prototype implementation.

6. Once the prototype is done and the results between the developer's prototype and tester's prototype match, work on the other requirement, and expand the prototype to final software.

7. When the testing approach was implemented, the following pros and cons regarding the testing approach were realized.

Pros of using the methodology:

Helps give a better understanding about the requirements.

Better design at the end of the cycle.

Reduced testing to be performed at the end of the cycle

Documents produced would be of higher quality.

Cons of using the methodology:

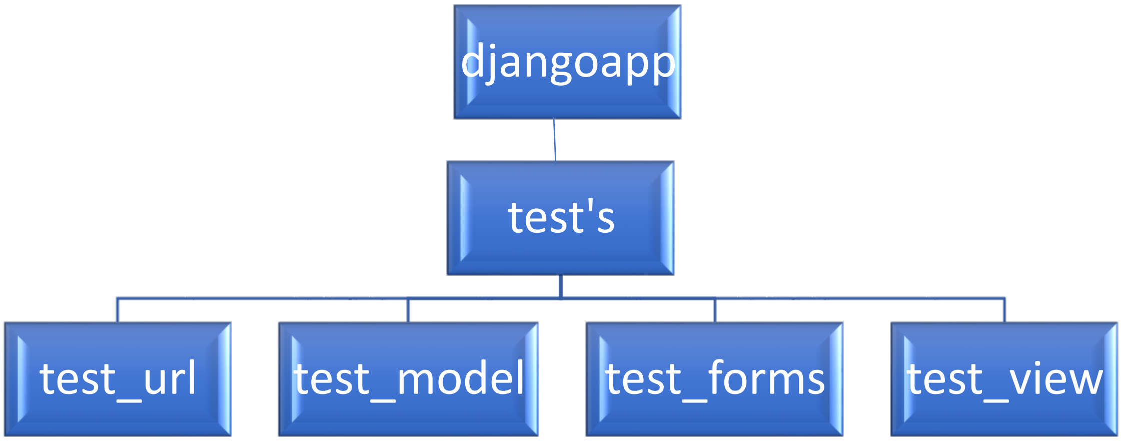
The person working on the document should be experienced.

There are increased time and money involved with testing.

Different viewpoints for the same problem can lead to varying results.

5.2. Interface Testing

This section will show the test-file hierarchy table, the test cases that were used to verify the interface table, and the results for the test-cases table.



5.3. Test Cases

Table shows the functional requirements used to write the test cases along with the test-case numbers for each test case and a short description of the test cases.

Test-Case Short Description	Test No.
To test the Login/Authentication interface for the Admin	TC001

To test the Login/Authentication interface for the user (Number of user)	TC0002
To test, Admin can view the information about all the users which are in website.	TC003
To test, Admin can add/delete items, user and branches.	TC004
External account can check summary page	TC005
To test, Admin can view the information about all the product with whole information about it.(number based on serial nomber)	TC006
To test that users can check all product in his branch.	TC007
To test that users are not able to do any update in other branches	TC008
External account cannot add or delete products	TC009

Table. List of Test Cases.

The following list includes the steps that should be taken by the user, the conditions that should be met for the successful execution of the test case, and the result that should be met for the test cases to pass.

TC01: To test the Login/Authentication interface

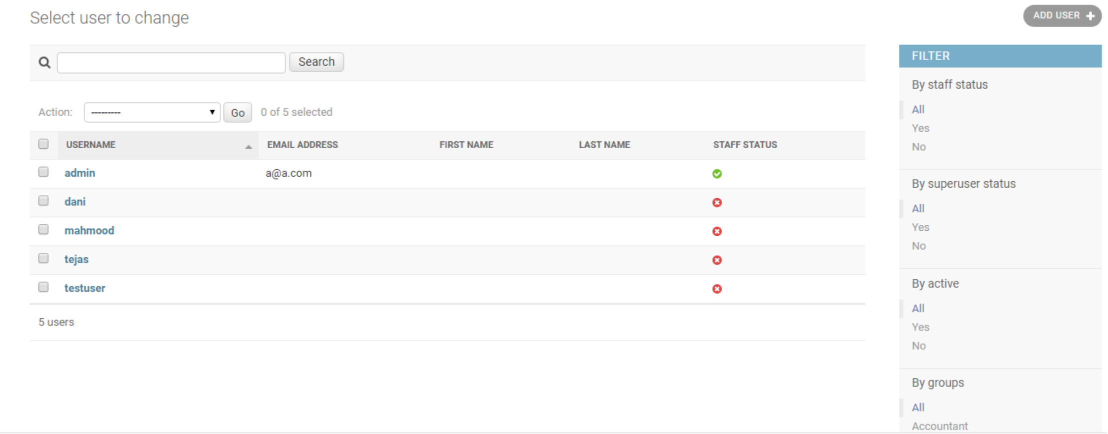
- Input: Username and Password
- Output: Valid Destination Page
- Valid Range: Username □Alphanumeric, Password □□Alphanumeric
- End Messages/Result
- ❖ If (User == Valid User), an order form appears to complete the checkout process.
- ❖ If (User!= Valid User), back to the Login interface.

TC02: To test the Login/Authentication interface for the user

- Input: Username and Password
- Output: Valid Destination Page

- Valid Range: User Name □Alphanumeric, Password □□Alphanumeric
- ❖ If (User == Valid User), an order form appears to complete the checkout process.
- ❖ If (User!= Valid User), back to the Login interface.

TC03: To test, Admin can view the information about all the users which are in website.
Description of Purpose: The system shows all the saved information from user's table.
□ Input
i. User Name □Alphanumeric, Password □□Alphanumeric
ii. User==Admin
iii. Selection==View Database
□ Output: User List
□ End messages/Result
i. If(login type == "Admin" & Database.clicked = 'true' and list.clicked=true and userlist.exists=true), then display users.
ii. If (login type == "Admin" &Database.clicked = 'true' and list.clicked=true and userlist.exists=false), then display the empty database.



Select user to change

0 of 5 selected

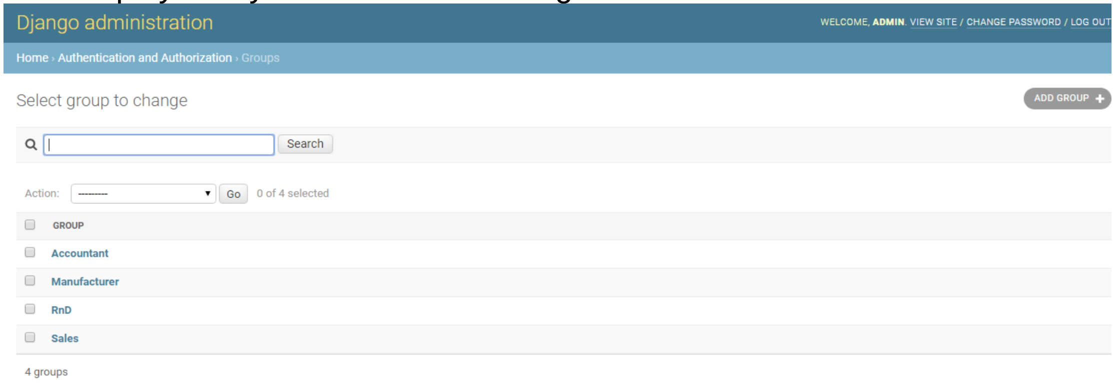
□	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
□	admin	a@a.com			✓
□	dani				✗
□	mahmood				✗
□	tejas				✗
□	testuser				✗

5 users

FILTER
By staff status
All
Yes
No
By superuser status
All
Yes
No
By active
All
Yes
No
By groups
All
Accountant

TC04: Admin can add/delete items and branches.
□ Description: The Admin can add or upload more items to a category or can add a completely new category. The Admin can also modify the price, information and status, etc. for the existing items and categories.
□ Input
i. User=Admin
ii. Selection=Items
iii. Selection=Categories
□ Output: New or modified items or categories in the shopping cart.
□ End messages/Result
i. If (User type = "Admin" &Selection = (Items || Category)&& Item/Category =existing), then display the modified items or categories in the dashboard.

If (User type == “Admin” &Selection == (Items || Category) & Item/ Category=existing), then display newly added items or categories in the dashboard.



Django administration

WELCOME, ADMIN. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home › Authentication and Authorization › Groups

Select group to change

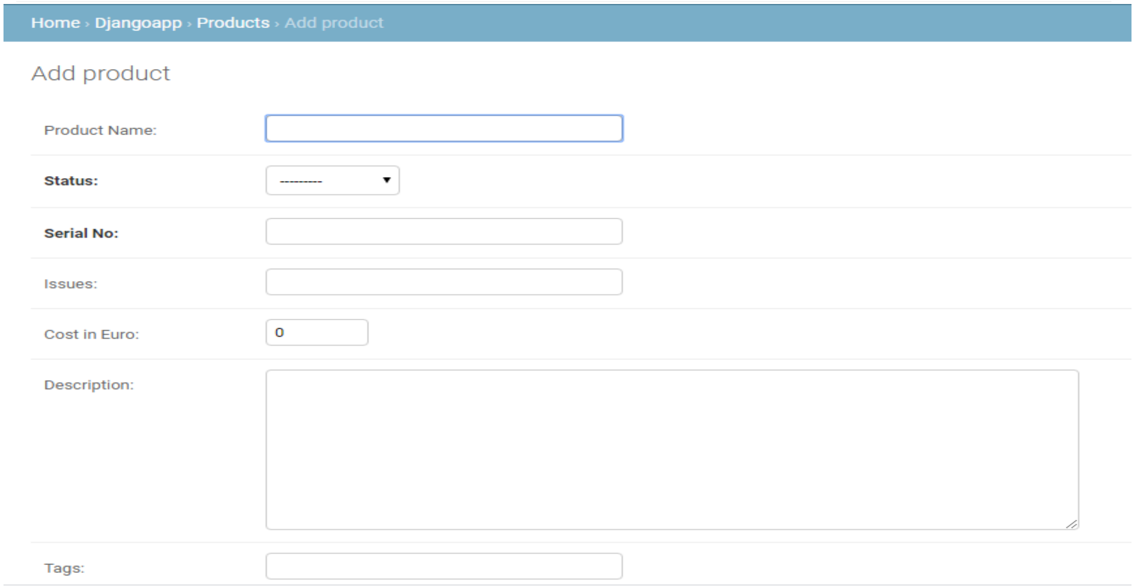
ADD GROUP +

Search

Action: --------- Go 0 of 4 selected

□ GROUP
□ Accountant
□ Manufacturer
□ RnD
□ Sales

4 groups



Home › Djangoapp › Products › Add product

Add product

Product Name:	
Status:	---------
Serial No:	
Issues:	
Cost in Euro:	0
Description:	
Tags:	

.

5. TC05: To test, External account can check summary page

□ Description: For the External account, a database, which contains all information about the products, is created after each user checks out the items and successfully places an order.

□ Input

i. User Name □Alphanumeric, Password □□Alphanumeric

ii. User== External account

Selection==Database product summary info

□ Output: A database with the products information with specific data.

i. If (login type == “External account” and and product list.exists=true), then display the database containing the product’s information.

ii. If (login type = “Admin” & and product list.exists=true), then will not display the database and refuse access.

iii. If (login type = “Employee” & product list.exists=true), then will not display the database and refuse access.

TC06: To test, Admin can view the information about all the product with whole information about it.

□ Description: The Admin can view all the products with all parameter in the system in the database.

□ Input

i. Username □Alphanumeric, Password □□Alphanumeric

ii. User==Admin

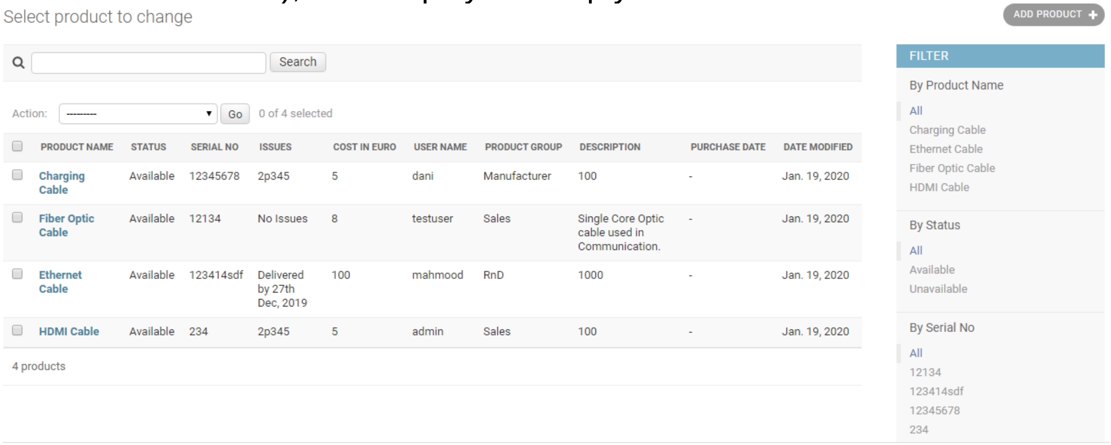
iii. Selection==View Database

□ Output: Product List

□ End messages/Result

i. If(login type == "Admin" & Database.clicked = 'true' and list.clicked=true and productlist.exists=true), then display users.

ii. If (login type == "Admin" &Database.clicked = 'true' and list.clicked=true and userlist.exists=false), then display the empty database.



Select product to change

□	PRODUCT NAME	STATUS	SERIAL NO	ISSUES	COST IN EURO	USER NAME	PRODUCT GROUP	DESCRIPTION	PURCHASE DATE	DATE MODIFIED
□	Charging Cable	Available	12345678	2p345	5	dani	Manufacturer	100	-	Jan. 19, 2020
□	Fiber Optic Cable	Available	12134	No Issues	8	testuser	Sales	Single Core Optic cable used in Communication.	-	Jan. 19, 2020
□	Ethernet Cable	Available	123414sdf	Delivered by 27th Dec, 2019	100	mahmood	RnD	1000	-	Jan. 19, 2020
□	HDMI Cable	Available	234	2p345	5	admin	Sales	100	-	Jan. 19, 2020

4 products

TC07: To test that Employee can check all product in his branch.

□ Description: The Employee cannot change any information on the other branch's he can only modify or delete in his branch

□ Input

i. Username □Alphanumeric, Password □□Alphanumeric

ii. User==Employee

iii. Selection==Edit or Delete

□ Output: Employee successfully or unsuccessfully based on this data in user table.

i. If (login type == "Employee" &Branch = 'true'), then display window Edit or Delete with all parameter related to product.

ii. If (login type == "Employee" & Branch = 'false') then no change in product parameter.

TC08: External accountant cannot add or delete products

Description: The External accountant cannot change any information on the product list

□ Input

i. Username □Alphanumeric, Password □□Alphanumeric

ii. User== External accountant

iii. Selection==Edit or Delete
 Output: External accountant can't change anything based on this data in user table.
If (login type == “External accountant” &Branch = ‘true’), not react from inventory

5.4. Results

This section lists the results that were produced by running the test cases. Table 3 lists the test cases that were used while testing the interface along with the expected result and the actual results for each test case.

Actual Result	Expected Result	Test Case Number
Pass	Pass	TC01
Pass	Pass	TC02
Pass	Pass	TC03
Pass	Pass	TC04
Pass	Pass	TC05
Pass	Pass	TC06
Pass	Pass	TC07

Table. List of Test-Case Results.

Chapter 6 Future update and Deviation:

In this chapter The Conclusion, Deviation and the Future Work for the software will be given. The recommend step's will not implement in this project phase, but it will be required if developer need upgrade this project.

6.1 Future update

1- If it finds that certain product is below the threshold, it will generate a purchase order for those item(s) and send it to the purchase department to buy it after approval

2- We also propose to include a special feature “Prediction”. This feature keeps track of any upcoming occasions, climatic changes and special events that may influence inventory needs for the upcoming week. The system will then predict the required resources for these events based on previously accumulated information/knowledge. It will now generate an updated purchase order in accordance with the predictions.

3- The product also aims to keep track of the shelf life of resources. If any resource nears the end of its shelf life, it would intimate to the manager (admin) the details of the quantity that is near its expiration date. The restaurant must function efficiently, the groceries must be tracked correctly, timely orders must be sent out to the vendors, and the inventory must always be maintained and updated.

4- Error Messages and Recovery Procedures:

- This section should list all the error messages. It should not simply repeat the error message but should also give a diagnosis and suggest recovery procedures.
- If recovery action is likely to involve loss of inputs or loss of stored data, the user should be reminded about backup and archiving procedures.
- Where all the details which this section would contain are included in comprehensive on-line “Help” files, the distributed version of the User Guide document may omit this section and include just a reference to the “Help” files. However, it may even then be found preferable to include all details in the User Guide document, at least while the system is under development.

5- Admin can view all the users logged in the system now.

6.2 Deviation:

1- Sign up page: the problem here is logic confusion because how any one can sign up to inventory without at least admin acceptance but in real the IT department (admin) create the account to new employee and assign him to specific group.

2- Anomous User: as mention before in sign up page the logic confusion there's the same here but in different way, what is the benefit of creating a user who does not have any authority or functionality on the website!

3- Upload Images to item parameter.

References

https://pypi.org/

https://www.python.org/

https://getbootstrap.com/

https://www.w3schools.com/

https://tutorial.djangogirls.org/en/