## TICKETING MANAGEMENT SYSTEM (whole)

<https://www.youtube.com/watch?v=xriCKexNT6s>

CHIDI OHIRI

# PROJET --

## Authentication

**User registration - authentication**

**All**

Ticket details

**Customer**

Create ticket

Update ticket

Active tickets

Closed tickets

**Engineer**

View ticket Queue

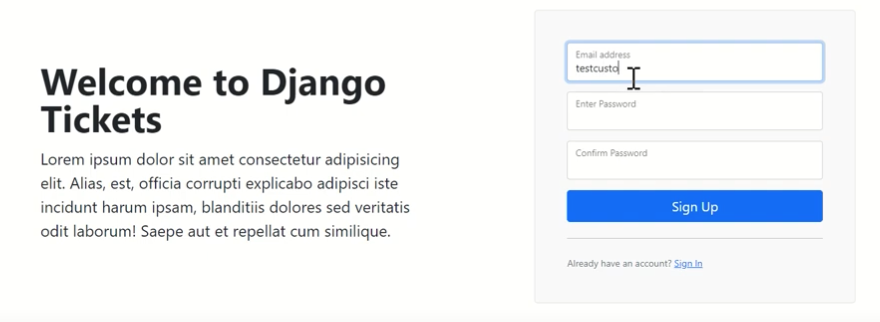
Accept ticket

Close ticket

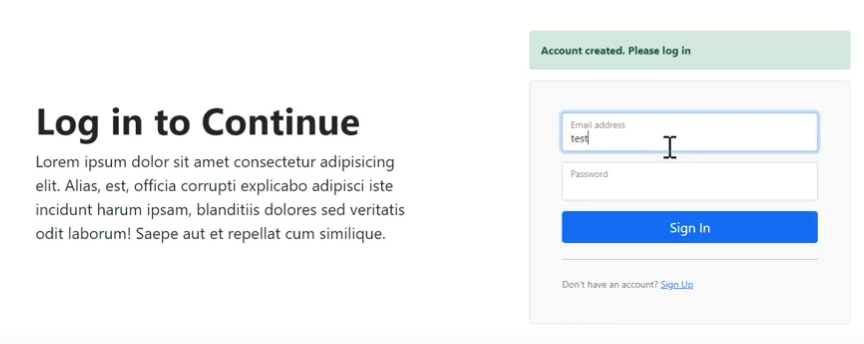
Active tickets

Closed tickets

SIGN UP



LOGIN



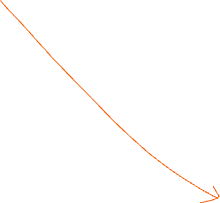
## CUSTOMER dashboard



Active, Closed and Total tickets

## 

## 



## Create ticket

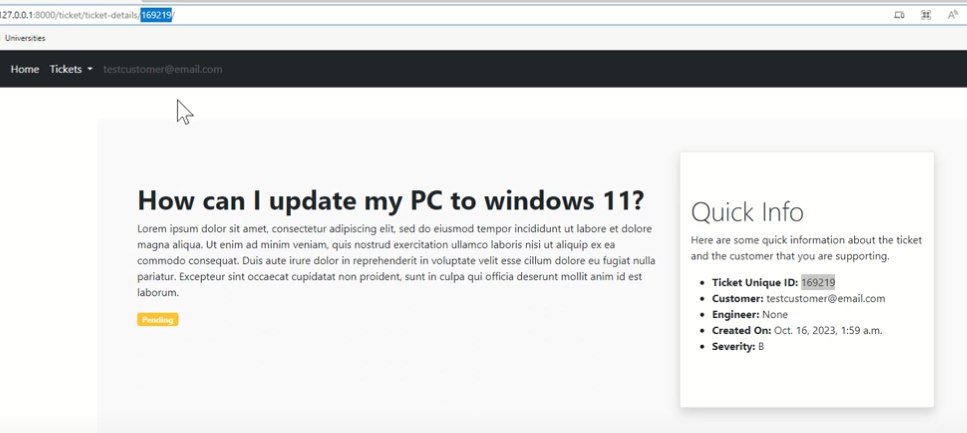


**Active tickets**

|  |  |
| --- | --- |
|  | **URL:** customer-active-tickets  Message  **Pending:** active ticket  **Created now**: time laps since the creation (humanize: ex. *35 second ago*) |

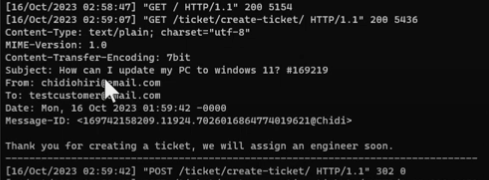
**View details:**





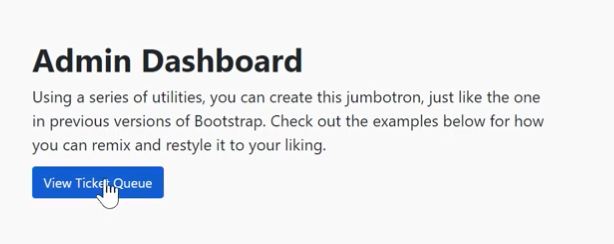
* Ticket has not been assigned to engineer yet
* Unique *ticket ID*generated from Django

Email sent to customer, while creating a ticket

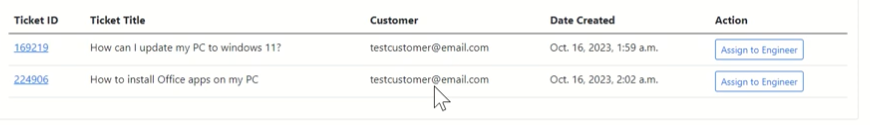


## Engineer dashboard



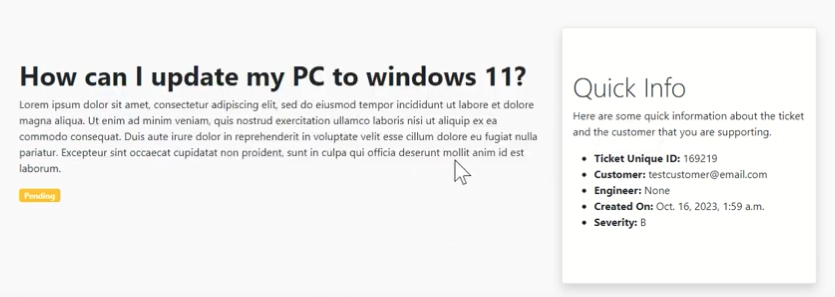
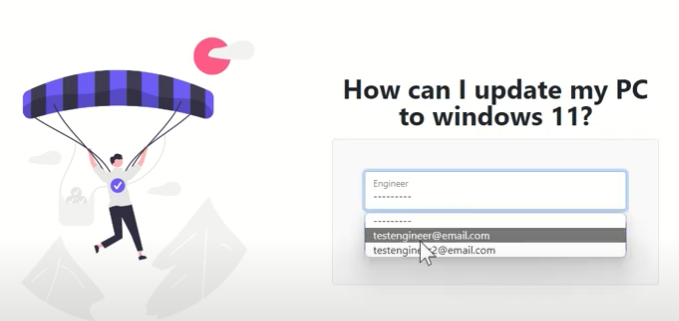




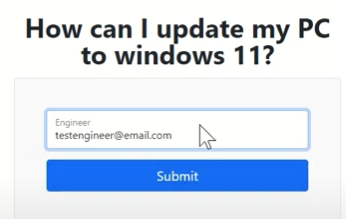




Ticket Details Assign to engineer

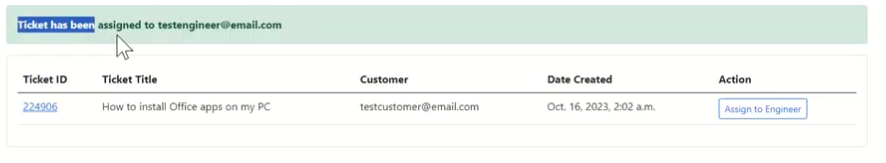




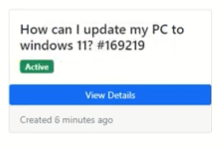


Ticket removed from the **admin/ticket-queue**

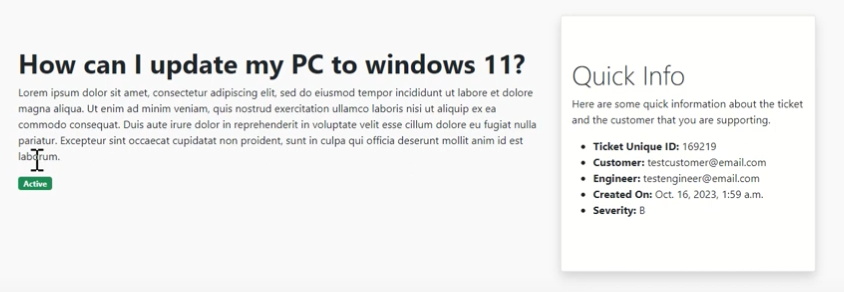




Ticket status become **Active** once assigned to an Engineer



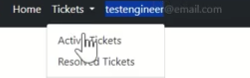
Change in ticket-details

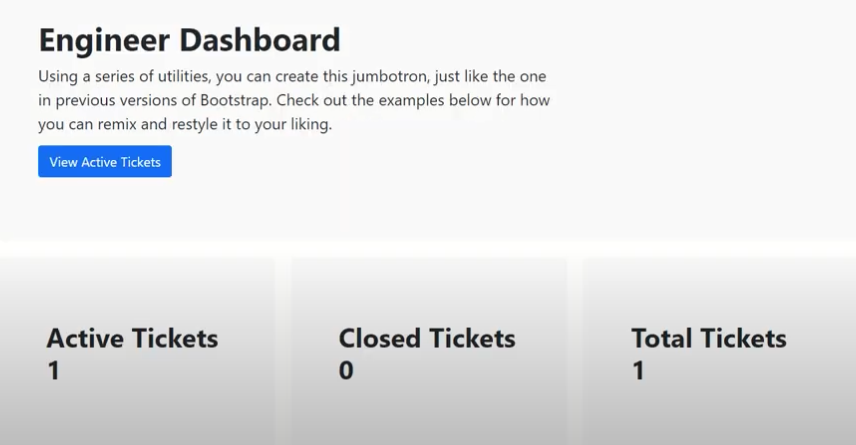




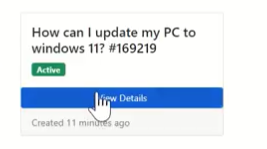
## ENGINEER dashboard







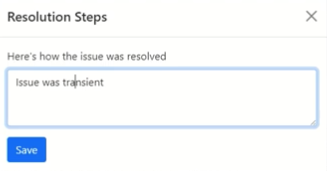
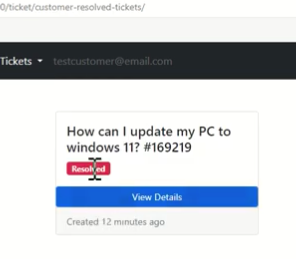
## Active tickets



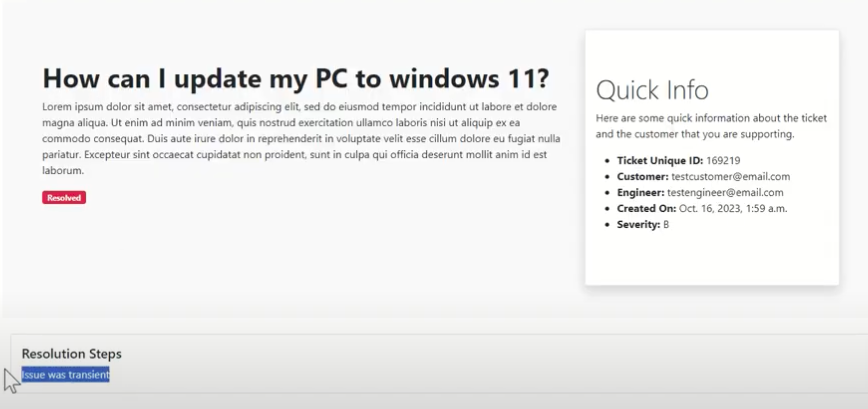












# PROJET --

# Create virtual env – install django

Prevent conflict with installed app on the global machine

pip install virtualenv

virtualenv env(environment name) source env/bin/activate

pip install Django

## Install new project

django-admin startproject django\_project

cd.. project name, open vs code

## Creating Apps

python3 manage.py startapp accounts

python3 manage.py startapp dashboard

python3 manage.py startapp ticket

python3 install django-widget-tweaks (form)

## Setting.py

import os

Put the 3 installed apps + tweaks

'ticket',

'dashboard',

'accounts',

'widget\_tweaks', *#form styling*

]

Tell Django that we’re going to use our own custom model

AUTH\_USER\_MODEL = 'accounts.User'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

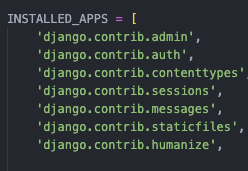
'DIRS': [

os.path.join(BASE\_DIR, 'templates/')

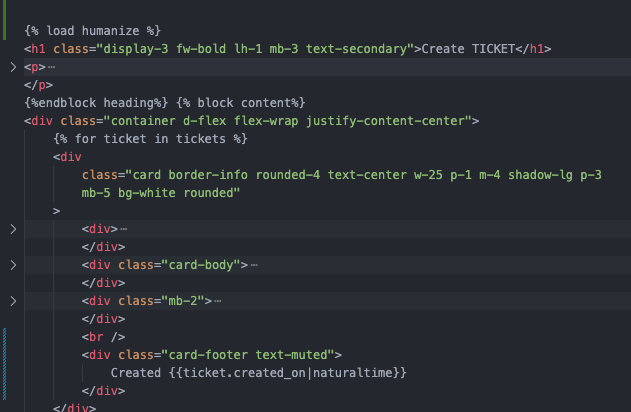
],

# HUMANISE

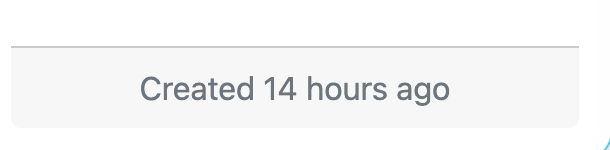
Setting



Tempalte



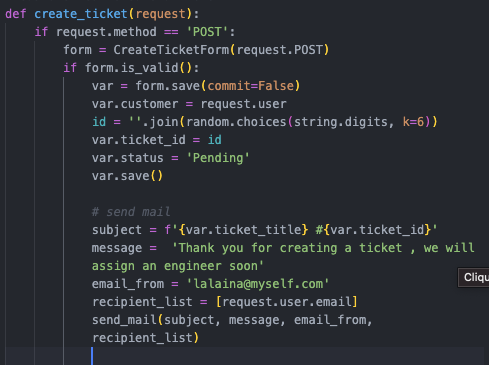




# Customer receives mail when he creates a ticket

Ticket/view

from django.core.mail import send\_mail



Setting.py



Simulation envoi mail sur le terminal