Title: A very brief introduction about installation of CMS (Contest Management System) on a (quite old) laptop running Ubuntu Desktop ver. 16.04

video:
https://drive.google.com/file/d/0B-lFexqGexwdMlR3aFVobTBvbkU/view?usp=share_link&resourcekey=0-YRXcMSc4qDLYcCat_cFhMg

Abstract: CMS, or Contest Management System, is a distributed system for running and organizing a programming contest.
This video-tutorial covers a very brief introduction of the installation of CMS (Contest Management System) and it's done primarily with the hope to serve as a starting point for the utilization of CMS for Educational purposes.
It is intentionally limited to the very basic steps, without ANY intention to cover all the troubleshootings you can find during the process on your own Computer or your Virtual Private Server or whatever other "architecture" you may utilize.
If you get stuck you can consider to join the cms-dev community on:
https://gitter.im/cms-dev/

You can find more information about CMS starting with a visit to the official web site:
https://cms-dev.github.io/
and the official docs:
https://cms.readthedocs.io/en/v1.3/

So Lets Start!

1. Installing Libraries:
First of all we need to install the required libraries.

...$ sudo apt-get update # Fetches the list of available updates

and then:

...$ sudo apt-get install build-essential openjdk-8-jre openjdk-8-jdk fpc postgresql postgresql-client gettext python2.7 iso-codes shared-mime-info stl-manual cgroup-lite

finally...

...$ sudo apt-get install python-setuptools python-tornado python-psycopg2 python-sqlalchemy python-psutil python-netifaces py thon-crypto python-tz python-six python-beautifulsoup python-mechanize python-coverage python-mock python-requests python-werkzeug python-gevent python-bcrypt python-chardet patool python-yaml python-sphinx python-cups

python-pypdf2 nginx-full

In this case the libraries where already installed in order to save time.
Usually the process will take several minutes.

2. Preparation steps:
Download latest stable version of CMS 1.3.rc0 from GitHub as an archive
https://github.com/cms-dev/cms/releases/download/v1.3.rc0/v1.3.rc0.tar.gz
then extract it on your filesystem in a directory of your choise.
...$ cd tutorial
~/tutorial$ tar -zxvf v1.3.rc0.tar.gz
You should then access the cms folder.
~/tutorial$ cd cms

In order to run CMS there are some preparation steps.
First we have to configure two files: cms.conf.sample and cms.ranking.conf.sample
(one for CMS itself and one for the rankings).
Samples for both files are in the directory: config/.
You have to copy them to the same file names but without the .sample suffix
(that is, to config/cms.conf and config/cms.ranking.conf) before modifying them.

~/tutorial/cms$cd config
~/tutorial/cms/config$cp cms.conf.sample cms.conf
~/tutorial/cms/config$cp cms.ranking.conf.sample cms.ranking.conf

We must change the connection string given in database; this usually means to change username, password and database name.
In order to keep this tutorial simple now we will modify only cms.conf with the terminal text editor vi.
~/tutorial/cms/config$vi cms.conf
we have to modify the line:
"database": "postgresql+psycopg2://cmsuser:your_password_here@localhost/cmsdb",
with our user: tutorial and password: xxxxx (of course change it with the username and password of your choise)
"database": "postgresql+psycopg2://tutorial:xxxxx@localhost/cmsdb",

:wq! to exit vi.

Now we have run the following command:
~/tutorial/cms$ sudo ./prerequisites.py install
This script will add you to the cmsuser group if you answer Y when asked.
Now you have to logout and login again to make the change effective.

3. Installing CMS
To install CMS on Ubuntu, you can issue:
~/tutorial/cms$ sudo python setup.py install

4. Configuring the DB
The first thing to do is to create the user and the database, for that you need to login as the postgres user first:
sudo su - postgres
Then, to create the user and the database, you need the following commands:
postgres$ createuser --username=postgres --pwprompt tutorial
postgres$ createdb --username=postgres --owner=tutorial cmsdb
postgres$ psql --username=postgres --dbname=cmsdb --command='ALTER SCHEMA public OWNER TO tutorial'
postgres$ psql --username=postgres --dbname=cmsdb --command='GRANT SELECT ON pg_largeobject TO tutorial'
postgres$ cmsInitDB
Now we can exit
postgres$ exit

5. Running CMS
Now you need an Admin account, which you need to create first using the AddAdmin command:
$ cmsAddAdmin name
CMS will create an admin account with username “name” and a random password that will be printed by the command. You can log in with this credentials, and then use the admin interface to modify the account or add other accounts.
The most immediate way to create a contest in CMS is using the admin interface. You can start the AdminWebServer using the command:
$ cmsAdminWebServer

You should now be able to start exploring the admin interface, by default at http://localhost:8889/. The interface is accessible with an admin account. Remember that we had just created one.
You can now create a contest giving a name and other details.

Now You can start CMS with the command:
$ cmsResourceService -a 1

Some cleaning:
 rm -rf /var/local/cache/cms/fs-cache-Worker-*/*
 rm -rf /tmp/tmp*