STEPS TO INSTALL AND CREATE DJANGO PROJECT

NOTE: The following commands were tested using the ubuntu environment so for windows or mac they might differ a bit.

INSTALLATION

1. Create a folder for the django project to be hosted.
2. Open it in the terminal
3. Configure a virtual environment
    - **virtualenv venv**
4. Activate the virtual environment.
    - **source venv/bin/activate**

   Note: to deactivate it is
    - **deactivate**
5. Install django (for the latest version)
    - **pip install django**
    - **sudo apt install python3-django**

   Note: to install a specific version you add a version like this:
    - **pip install django==3.0.8**
6. Create django project
    - **django-admin startproject** <name of the project>
      Like
        - **Django-admin startproject spareParts**
7. Go into the project
    - **cd spareParts**
8. Migrate
    - **python3 manage.py migrate**
9. Create superuser
    - **python3 manage.py createsuperuser**

   Enter username like:
    - **spareMotors**

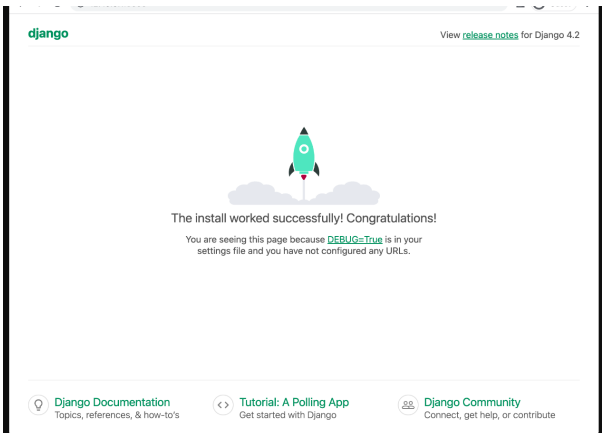
   Enter email: this is optional

   Enter password : **……..**
10. Create/start django application
    - **python3 manage.py startapp** <app name>
      Like
        - **Python3 manage.py startapp spareApp**

    Note: A Django project can have one or more applications.

11. Open your folder into your code editor like visual studio code
    - **code .**
12. Run the server from your terminal to confirm that everything works perfectly.

    Note: If it is successful it will display a Django project with an Aeroplane like cartoon.



Note: we run the server from the project folder

- **python3 manage.py runserver**
  Note: incase you need to specify a port number
  - **python3 manage.py runserver 8001**

13. Then go ahead and do other installations to be used
    - Crispy forms
      - **pip install django-crispy-forms**

      Note: This helps in creating forms in our project.
    - Bootstrap5/ crispy bootstrap
      - **pip install django-bootstrap-v5**
        or
      - **Pip install crispy-bootstrap4**

      Note: This helps in displaying crispy forms
    - Filters
      - **pip install django-filter**

      Note: These filters help to activate the search fields.
14.

Settings adjustments.

1. Go into your settings file <**settings.py**>
2. Install the application into the installed
   - **'spareApp',**
3. Append the downloads
   a. Crispy forms

   ```
   'crispy_forms',
   ```

   b. Bootstrap5

   ```
   'crispy_bootstrap4',
   ```

   c. Django filters

   ```
   'django_filters',
   ```

And create the base pack for crispy forms

```
CRISPY_TEMPLATE_PACK ='bootstrap4'
```

4. Create the **templates** folder into your project and open templates files like index.html
   And then
   Register your templates path into the settings file under

```
  'DIRS': [os.path.join(BASE_DIR,'templates')],
```

   Note: The templates folder is where all the **templates** are created i.e known as(**html files**)

5. Create the static folder into your application and open static files likes style.css
   And then
   Register the static path into the settings file

```
STATIC_ROOT = os.path.join(BASE_DIR,'static')
STATICFILES_DIR = [os.path.join(BASE_DIR,'spareApp/static')]
```

   Note: static folder takes in all css files, images and javascript files

6. Make the redirections for login and logout.

```
LOGIN_REDIRECT_URL = 'home'
LOGOUT_REDIRECT_URL = 'logout'
```

Configuring templates and static folders

7. Go to your **urls.py file** in you project
   First **import include** from **django.urls**

```
from django.urls import path,include
```

   And then
   Create an include pattern under **urlpatterns**

```
path('',include('spareApp.urls')),
```

   Note: this directs all other urls to the app's urls

8. Go to your app and create a **urls.py file**
   import path from django.url

```
from django.urls import path
```

   Import views from your app

```
from spareApp import views
```

   And then create the path for other views
   Like for **index** and **home**

```
path('', views.index, name= 'index'),
path('home', views.home, name= 'home'),
```

9. In your **views.py file** in your app create the views

```
def index(request):
return render(request, 'spare/index.html')
```

Note: views are also known as
functions in structured programming
Methods in object oriented programming

Configuring models

Note: Models help us to create tables where our data is going to be stored in the database.

10. Import models from django

```
from django.db import models
```

11. Create classes/tables

```
class Product(models.Model):

Category_name = models.ForeignKey(Category,
on_delete= models.CASCADE, null=False, blank=
False)
item_name = models.CharField(max_length=50, null=
False, blank= False)
total_quantity = models.IntegerField(default= 0,
null=False, blank=False,
validators=[MinValueValidator(1)])

def __str__(self):
return self.item_name
```

12.