1. **Create Project folder and then navigate to project folder**

cd FinalProjectKiwiFeed

1. **Create Virtual Environment**

Environment name :env

Create env with same name so that in git ignore file env folder will not be shared with team

python -m venv env

1. **Activate Virtual Environment**

In command line : env\Scripts\Activate

1. **Create ‘.gitignore’ file**

inside the Project folder where ‘env’ folder exits

Create file name ‘.gitignore’ with vs code

Add folder name inside ‘.gitignore’ file

env/

.env/

add and commit it to your Git repository

go to make repository folder and open command line and make this commit

1. git add .gitignore

git commit -m .gitignore

or

you can use github desktop application too for make this commit

1. **Install Django and other Library**

inside the Project folder where ‘env’ folder exits

pip install Django

1. **Create requirement.txt file**

Go to main project folder where environment folder –‘env’ exits

Activate environment

Update pip by

pip install --upgrade pip

or

python.exe -m pip install --upgrade pip

create or update requirement.txt file

pip freeze > requirements.txt

1. **Update packages from requirements.txt file**

Go to main project folder where environment folder –‘env’ exits

Activate environment

Install packages from requirements.txt file

pip install -r requirements.txt

1. **Create Django Project kiwiFeeds**

django-admin startproject project\_kiwifeeds

1. **Navigate to Project Folder project\_kiwifeeds**

Cd project\_kiwifeeds

1. **Create Django Apps**

python manage.py startapp app\_restaurant

in setting.py add app in installed app

INSTALLED\_APPS = [

   …

    'app\_restaurant',

]

1. **Create utility folder in main project folder**

static

static/defaultImages

media

media/restaurant/menuItem

media/restaurant/resturantImages

media/user

templates

templates/restaurant

templates/user

templates/base

templates/components

1. **Setup template Folder**

Change in setting.py page of project

import os

TEMPLATES = [

{

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

'DIRS': [os.path.join(BASE\_DIR, 'templates')],

# ...

},

]

1. **Setup static Folder**

Define the path to your static root directory in setting.py file

STATICFILES\_DIRS=[BASE\_DIR / "static"]

STATIC\_ROOT = BASE\_DIR / 'staticfiles'

Add url.py file

from django.conf import settings

from django.conf.urls.static import static

if settings.DEBUG:

urlpatterns += static(settings.STATIC\_URL, document\_root=settings.STATIC\_ROOT)

and in each html page where we need to use images from static folder use tag

{% load static %}

1. **Setup media Folder**

Define the path to your media root directory in setting.py file

# media url and path setting

MEDIA\_URL = '/media/'

MEDIA\_ROOT = os.path.join(BASE\_DIR, 'media')

Add url.py file

if settings.DEBUG:

    urlpatterns += static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

to use media folder setting to upload file , add following code to model.py

class YourModel(models.Model):

image = models.ImageField(upload\_to='images/folderPath/')

1. **Add extra library**

Go to you main project folder where env folder exits

Install pillow for image

pip install Pillow

and then update requirements.txt file

pip freeze > requirements.txt

1. **Django-Debug Tool Setup**

Follow instruction:

<https://django-debug-toolbar.readthedocs.io/en/latest/installation.html>

and then update requirements.txt file

pip freeze > requirements.txt

1. **Setup Tailwind**

Open terminal where ‘env’ file exits

Activate environment

env\Scripts\Activate

Then install tailwind

python -m pip install django-tailwind

add tailwind in setting.py file the Django project:

INSTALLED\_APPS = [

# other Django apps

'tailwind',

]

Create a *Tailwind CSS* compatible *Django* app by the name of ‘theme’

Navigate to project folder where manage.py file exits

python manage.py tailwind init

Tailwind application 'theme' has been successfully created.

Please add 'theme' to INSTALLED\_APPS in settings.py,

INSTALLED\_APPS = [

# other Django apps

‘theme’,

]

Register the generated 'theme' app by adding the following line to settings.py file:

TAILWIND\_APP\_NAME = 'theme'

Check node and npm version by command line

node -v

npm -v

if both node and npm exits then add npm path to your setting.py file

my system path is mentioned below , you have to find out yours if getting error

NPM\_BIN\_PATH=r"C:\Program Files\nodejs\npm.cmd"

then run the following command to install Tailwind CSS dependencies:

python manage.py tailwind install

to use tailwind ,add tailwind\_tags to base.html pages

{% load static tailwind\_tags %}

...

<head>

...

{% tailwind\_css %}

...

</head>

After all setup To use tailwind in project

Open command line in project folder

Navigate to project folder where manage.py exits

Initialize tailwind by following command

python manage.py tailwind start

run this terminal along with

python manage.py runserver

or

follow instruction:

<https://django-tailwind.readthedocs.io/en/latest/installation.html>

and then update requirements.txt file

pip freeze > requirements.txt

1. Install django\_browser\_reload

Open command line and activate environment

Install djnago browser reload package

pip install django\_browser\_reload

Add django\_browser\_reload to INSTALLED\_APPS in settings.py,

INSTALLED\_APPS = [

# other Django apps

‘django\_browser\_reload’,

]

Add 'browser\_reload.middleware.BrowserReloadMiddleware' to MIDDLEWARE:

In the same settings**.**py file, add

MIDDLEWARE = [

# ...

"django\_browser\_reload.middleware.BrowserReloadMiddleware",

# ...

]

Include django\_browser\_reload URL in your root url.py:

from django.urls import include, path

urlpatterns = [

...,

path("\_\_reload\_\_/", include("django\_browser\_reload.urls")),

]

and then update requirements.txt file

pip freeze > requirements.txt