`Django Python

Node js vs Django

To explain simply, if you are building a car Node js provides you tools to make parts from scratch and assemble the parts, whereas Django is like a well stablished car manufacture company with all the necessary parts which have manufacturing machines that helps you to assemble car.

First, lets learn to create all necessary folders and connect them.

# Starting a project

```
django-admin startproject demo_site
```

Select folder where you want to start new project and run above code on Vs Code terminal This automatically creates folder (demo_site) and file(manage.py) and this is where you will be performing all the action throughout the project. This is also called project level file.

# Loading project on Live Server

```
python manage.py runserver
```

If you want to run project on customized port try:

```
python manage.py runserver 8080
```

# Creating app folder

```
python manage.py startapp     landing_app
```

This will create number of .py files automatically, among them view.py file is the place where we write all the stuffs that we want to display on our webpage.
You have to manually create urls.py file within app level folder and provide links of view.py and connect our app level to project.

In app level: Creating view and connecting it with url

view.py

```
from django.http import HttpResponse
def index(request):
        return HttpResponse("Hello World!")
```

urls.py

```
from django.urls import path
form . import views
```

```
urlpatterns =[
        path(' ',views.index,name='index')]
```

In project level: Connecting app level urls with project level urls

urls.py

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


urlpatterns =[
        path('demo_app/',include('demo_app.urls'),
        path('admin/',admin.site.urls), ]
```

views.py

```
from django.http.response import HttpResponse
def homepage_view(request):
        return HttpResponse("This content will be displayed on home page url!")
```

#DYNAMIC VIEWS & LOGIC

For example, suppose we are creating one magazine and we want separate view.py for sport, finance, politics and others and link all the views.py to our app level urls.py

Doing these manually line by line would violate DRY principle so we introduce Dynamic way

App level:
view.py

```
article={
        'sports':'This will be shown in url/demo_app/sports',
        'finance':'This will be in url/demo_app/finance',
        'politice':'xxxxxxyyyyyyzzzzzzzz',
        'others':'page for others' }

def news_view(request,topic):
        return HttpResponse(article[topic])

def addition_view(request,num1,num2):
        sum-num1+num2
        result = f"{num1}+{num2}={sum}"
```

```
    return HttpResponse(str(result))
```

urls.py

```
from django.py import path
form . import views

urlpatterns=[
    path('<str:topic>/',views.news_view),
    path('<int:num1>/ <int:num2>', views.addition_view)]
```