* Day 2 :

→ python 3.7.2

→ website : last 5th executable installer (Add path ☑)

→ cmd : pip install django

→ create folder "DJANGO"   after this cdmyproject

→ create project folder by cmd : django-admin startproject <projname>

→ create App folder by cmd : python manage.py startapp <app-name>

⊛ → In url.py of myproject :

• from myapp import views

• In urlpatterns create path

→ path('flipkart', views. flipkart),

• then

⊛ → In views.py of myapp (   )

• from django.http import HttpResponse

create function

• def flipkart (request) :
        return HttpResponse ("hello welcome to flipkart")

→ then (new terminal)

• write command : python manage.py runserver

→ link will come : http://127.0.0.1:8000/

• ctrl+ click ↑ to open

→ Then put /flipkart in the above link

→ Successfully completed

* Day 3 : templat rendering & passing views data into template

→ Open Django folder

→ Open myproject

(☆) → create folder " templates "

→ then from VScode : template [□ New File...]

(✴) → name new file as sample.html

→ ! : abbreviation of html body

→ <body>

 <h1> Welcome to my HTML website <h1>

 <body>

(✴) → then open urls.py of myproject

→ In urlpatterns = [   ]

→ create a new path

 * path ('', views.fun 1) , ←

(✴) → then open views.py of myapp

→ create new function

(" def fun1 (request):

 return render (request ,'sample.html')

→ In new terminal " python manage.py runserver "

→ Open link 127.0.0.1 : 8000

 → Successful ←

* NEXT TASK : MOBILE

(✴) In views.py of myapp : create new function

 def mobile (request) :

 data = { 'mobile1' : 'samsung' ,

 'mobile2' : 'apple' }

 return render (request ,'mobile.html', data)

(✴) In urls.py of myproject

 urlpatterns = [ ] create new path

 path ('mobile', views.mobile)

(✴) In settings.py of myproject

 import os

 TEMPLATES = [{

 'DIRS' : [os.path.join

 (BASE_DIR, 'templates')]

(✴) In template folder. create new file mobile.html

 <h1> welcome to my mobile website :

 {{mobile1}} and {{mobile2}} <h1 >

*Day 4 : STATIC FILES

⊛
- → Download WinRAR 6.23
- → Download Templati from "bootstrapmade.com"
- → After templati download complition click & open with winRar.
- → Then iPortfolio → copy index file
- → paste this index file in templates folder

⊛
- → In views.py create a function

  def index (request):

      return render (request, 'index.html')

⊛
- → In urls.py create a path

  path ('index', views.index)

⊛
- → Open iPortfolio copy "assets" file paste in Static
- → Create a new folder "Static"

⊛ → Open index.html, In line3 write {% load static %}

⊛ → In index.html
  wherever line starts with assets (from line 13)

  {% static 'assets/..../...png' %}

  [Line 13 14, 19 to 24, 27, 646 to 654, 657]

⊛ → In Settings.py :

  line 118 djangoproject.com (ctrl+click) open & copy STATICFILES_DIR

  paste that DIR code under STATIC_URL = '/static/'

⊛ → New Terminal : python manag.py runserver 8002 → [If you want to change
  link appears : (ctrl+click it)   last 4 digits of link]
  chrome : link / index

  . successfull .

Date / /

* Day-4 * Aug-25-23   * Template Inheritance *

, from cmd (by commands refer 1st page)

* Optional : new myproject , new myapp , make new templates
    folder in myproject2

⊕ Create file grocery.html file in templates & write code :
    `<h1> this is my nav bar <h1>`

    `<h1> this is my main content <h1>`

    `<h1> this is my footer <h1>`

⊕ create file common.html file in templates
    `<h1> this is my nav bar <h1>`

    : ← main content (after 6 steps below)(next page)
    :

    `<h1> this is my footer <h1>`

⊕ Then open grocery.html & clear (Ctrl+A) & Backspace
    & Instead of writing just extend

    `{% extends 'common.html' %}` = qinger format

    `<h1> main content lorem 10 <h1>` • lorem = generates random
                                                        text.

⊕ In views.py of ~~migrations~~ myapp & create function :
    def gro (request):
        return render (request , 'grocery.html')

⊕ In urls.py of myproject2 , create path in URL patterns
    path (' ' , views .gro). &

    Import [views by : from myapp import views]

⊕ In settings.py of myproject2 :
    import os   &

    In templates in DIRS write :
    'DIRS' : [os.path.join (BASE_DIR , 'templates')]

⊕ Open grocery.html write block tags to inherit
    `{% extends 'common.html' %}`
    `{% block content %}`
        `<h1> ̅ <h1>`
        main lorem
    `{% end block %}`

✳ block content holds the main content & we are writing these tags to inherit the main content (lorem 10) from qoury.html to common.html

✳ Open common.html & in the space (given in middle of nav & footer) : write block tags to derive the main content from qoury.html

{% block content %}

{% endblock %}

✳ Open terminal : python manage.py runserver

→ Website Appears ←

✳ Then go back to CMD which already has commands for creating project & app & code. at its last line

→ write new command: python manage.py migrate   OK OK OK OK

after this command db.sqlite3 size changes from 0kb to 136kb

✳ In models.py ~~of migrations~~ myapp ; create model for product

class Product (models.Model):

   name = models.CharField (max_length = 20)

   price = models.IntegerField (max_length = 10)

   def __str__ (self):
      return self.name   (* name function is used to display product names like phone, watch instead of p1, p2)

✳ In settings.py of myproject2 ; In INSTALLED_APPS = [
   'myapp'] ; create myapp in this ↗ (App's name)

✳ Open new Terminal & type : python manage.py makemigrations

   OP : create model Product

⊛ Open 0001_initial.py of migrations in myapp &
see we got schema generated.

⊛ Open Terminal : & write : To apply schema into database
python manage.py migrate   ok
                                    ok
                                    ok

⊛ Now, In terminal ; write :

python manage.py createsuperuser
Id Rizwanishere
Email : press Enter
Password : Rizwan123

⊛ Again migrate & makemigrations commands.

⊛ In admin.py of my myapp write :
admin.site.register (Product) &
Import using : from .models import Product
                    ↳ (used coz models & admin belong to same folder)

⊛ In terminal : runserver : link appears (ctrl+click)
now /admin after link
  → Django administration appears
  → login by username & password.
  → In myapp you can add products
       & can do CRUD (create, update, Delete Operations)