Django

- is free & open source web framework. it is written in ~~Pthy~~ Python
- it follows the model-view-template (MVT) architecture pattern
- It is maintained by Django software

---

feature of Django

1.) fast 2.) fully loaded 3.) Security
4.) Scalability

Verision

```
Pip install Django == 4.1.1
```

Pip → Python install package

How to create a folder

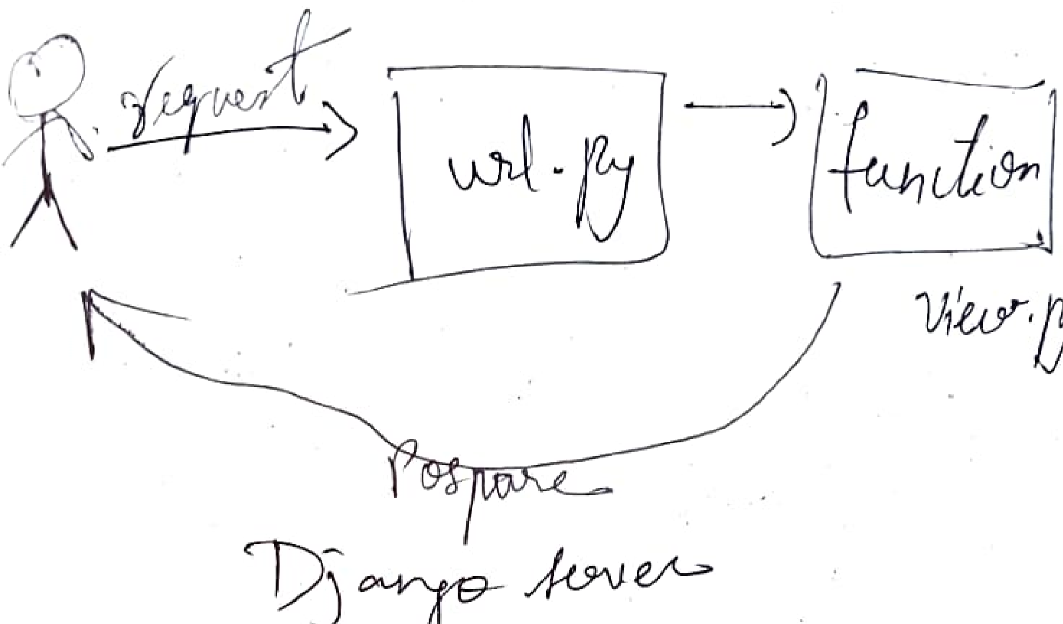
```
⇒ django-admin startproject newproject
⇒ cd newproject
```

How to create a application

```
⇒ Python manage.py startapp adminy
```

⇒ Python manage.py runserver

setting.py } all the paths are there



In views.py file

```
from django.shortcuts import render
from django.http import HttpResponse
# Create your views here
def display(request):
    s = '<h1> the django</h>'
    return HttpResponse(s)
```

In Urls.py

```
from django.contrib import admin
from django.urls import path
from admin import views
urlpatterns = [

        Path('admin /', admin.site.urls),
        Path('v/', ~~admin~~ veius.display)
    ]
```

In setting.py.

In installed apps.
add ("admin",
  }
)

$$\begin{tikzcd}
 & \text{MVT} \arrow[dl] \arrow[d] \arrow[dr] & \\
\text{model} \arrow[d] & \text{View} \arrow[d] & \text{Template} \arrow[d] \\
\text{database} & \text{business logic} & \text{frontend} \arrow[d] \\
 & & \text{HTML, CSS}
\end{tikzcd}$$

Django is a Framework

Create new folder in New project folder
→ templates
add files like index.html.

remove httpresponse from views.py

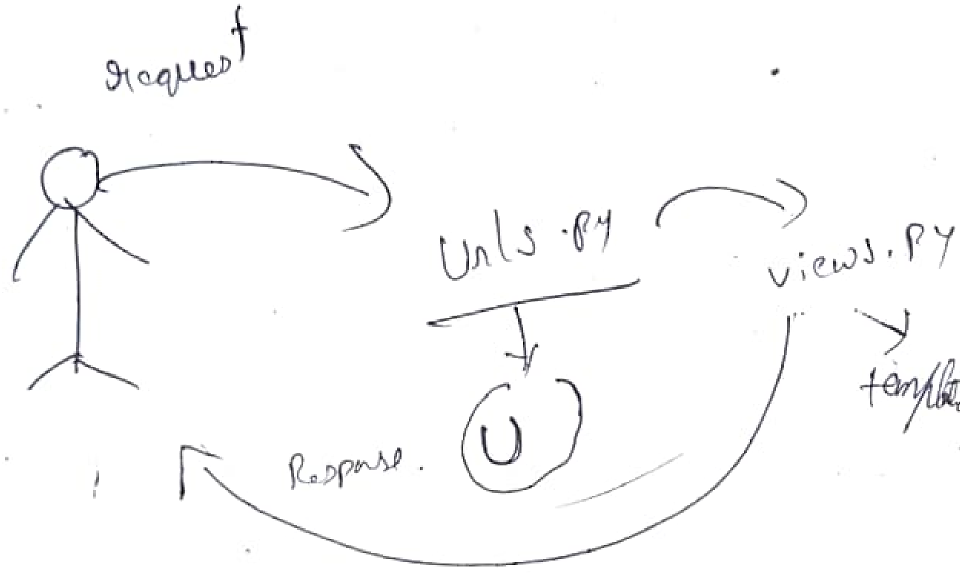
In views.py.

from django.shortcuts import render

def fun(request):
intended → return render(request, 'index.html').

In settings.py
add → import os
In Templates = [.
'DIRS': [os.path.join(BASE_DIR, 'templates')]

In templates → index.html
<!Doctype html>
<body>
<h1>hello this is my template</h1>.
</body>



Templates to render
3AM HTml

```
<!DOCTYE >

<body>
    <h1>this is templt</h1>
<h1> my name is {{ name }} age is {{age}} </h1>
</body>.
```

views.py :

```
from django.shortcuts import render
def fun(request):
    data={
        'name': 'Zia',    return render(request,'sum.html',
        'age' : 20                              data)
    }
```

Bootstrapmade.com → templates → free models

Download hue baad extract, then open the folder & only copy the index.html & paste it in our myproject index.html (education ka [illegible])

assets ka folder copy karlo

Create a new folder called static i
Beside manage.py & admin.

Paste assets folder inside static folder

Phir - setting.py me aao
# static file ki link click karo
120. STATIC_URL = 'static/'
121. yaha par paste karo

[staticfiles_DIRS = [
    Base_DIR / "static",

then in index.html.
in 2 line
{% load static %}

Use jinger format

In 14 line
<link href="{% static 'assets/img/favicon.png' %}

Trick.
select karo line phir alt hold karke
Pura select karo dusre lines ~~pir~~ start
editing lines for eg. like that.

Url.py
'Ye add Karna hai' Path('index/', views.index)

In Views -
def index(request):
    return render(request, 'index.html')

Next topic.

Block method.

from index.html copy all the code & emty the file then create new common ~~sample~~.html in templats & paste the code & remove all the thing except header & footer.

---

then In index.html

```
{% extends 'comenon.html' %}
{% block content %}
<h1> hello this is my index </h1>
{% endblock %}
```

In common.html

90 line
```
<main id="main">
    {% block content %}
    {% endblock %}
```

include method.

In Url ih
add Path('index1', views. index1)

```
{% include ~~%~~ 'footer.html' %}
```

In index1.html

```
{% include 'header.html' %}
 <h1> this is my content </h1>
{% include 'footer.html' %}
```

• create project named models & app as modelapp

Models
we can access LE

In models.py

```
from django.db import models

class Employee (models.Model):
    eno = models.Integerfield ()
    ename = models.~~Char~~field (Max_Length=35)
    esal = models.Floatfield ( )
    eaddr = models.charfield (max_length=35)
```

In Admin.Py

```
from django.contrib import admin
from modelapp.models import Employee

admin.site.register (Employee)
```

Insetting.Py
import os
add in install apps
'modelapp',
}

In Cmd type

→ python manage.py migrate
(OK
OK)

→ python manage.py makemigrations
(- create model employee

→ Python manage.py createsuperuser

Enter Username
Email & Password.

---

then python manage.py runserver
then /admin
enter ~~email &~~ Username & password

```
import pandas
dataframe= pandas.read_csv('studa  )
dataframe = dataframe.drop('Timestamp', axis=1)
dataframe =
dataframe= dataframe.drop(['Username', mobileno'
                                                  ,)
dataframe

dataframe.isna().sum() #isna=isnull() attri

dataframe.fillna(dataframe.mean())
dataframe.info()
from sklearn.preprocessing import LabelEncode
Le = LabelEncoder()
dataframe['Gender']= le.fit_transform
                              (dataframe['Gende
dataframe

dataframe['Branch'] = Le.fit_transform(dataframe
                                        ['Branch'])
dataframe('Result']= Le.fit_transform(dataframe
                                      ['Result']
-> dataframe
```

~~Google colab~~

~~x & y~~

~~[x~~

~~[y~~

~~from sklearn.model_selection~~

```
x = dataframe.iloc[:,0:-1]
x.isna().sum()   # functio to check the
                    null values

x = x.fillna(x.mean()) # fillna Used to fill the
                          null value which is
                               present
y = datframe.iloc[:,-1]
y
```

---

day #2

```
from sklearn.model_selection import train_test
                                       _split

x_train, x-test, y_train, y_test = train_test_split
                          (x, y, test_size=0.2)

x_train
x_test
y_train
y_test
```

```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier( )
rfc.fit (x_train, y_train)
y_pred = rfc.predict (X_test) # to test the result
y_pred
from ~~sk~~ sklearn.metrics import confusion_matrix
~~cm~~
cm = confusion_matrix (y_test, y_pred)
cm
```

```
import seaborn as sns
from matplotlib import pyplot as plt
sns.heatmap(cm,
            annot = True,
            xticklabels = ['pass', 'fail'],
            yticklabels = ['pass', 'fail'])
plt.ylabel('Prediction', fontsize=13)
plt.xlabel('Actual', fontsize = 13)
plt.title('Confusion Matrix', fontsize=17)
plt.show( )
```

full line

shift

#3 day (Monday).

Continue

```
from sklearn.metrice import accuracy_score
acc1 = accuracy_score (y_test, y_pred)

acc1

from sklearn.tree import DecisionTreeClassifier
# algorithm
dtc = DecisionTreeClassifier ()

dtc.fit (x_train, y_train)
y_pred = dtc.predict(x_test)
from sklearn.metrics import confusion_matrix
cm = confusion_matrix (y_test, y_pred)
cm

from sklearn.metrice import accuracy_score
acc2 = accuracy_score (y_test,y_pred)
acc2
```

Random forest Regressor [Wine ka data se karna hai

```
import sklearn.model_selection train_test_split
x_train, x_test, y_train, y_test = train_test_split
                                   (x, y, test_size = 0.3)

x_train
x_test
y_train
y_test

from sklearn.ensemble import RandomForestRegressor
rfr = RandomForestRegressor().
rfr.fit(x_train, y_train)
y_pred = rfr.predict(x_test)
from sklearn.metrics import ~~confusion_matrix~~
                            mean_absolute_error,
                            mean_squared_error

mae = mean_absolute_error(y_test, y_pred)
mae

mse = mean_squared_error(y_test, y_pred)
mse

from sklearn.linear_model import LinearRegression
 # Algorithm
lr = LinearRegression()
lr.fit(x_train, y_train)
```

```
import pandas as pd
dataset = pd. read_csv( (wine) )
dataset
dataset . isna ( ). sum( )
dataset . info ( )
x = dataset . iloc[ :,:-1]
y = dataset . iloc [ :, -1]
x
y
```

starting one Karo

```
y_pred = lr. predict ( x_test)
mae = mean_absolute_error (y_test, y_pred)
mae
mse = mean_squared_error (y_test, y_pred)
mse
```

(Pip install jupyter notebook) cmd,

test_set = 'MNIST-JPG-testing'

2 din nai aye open

https://shortusl.at/ivGLW) - Training Jupyter notebook

(https://shorturl.at/cgsy4) - trained model

https://shorturl.at/fiXY2

```
from keras.models import load_model
trained model = load_model('digit_recog_87epoch.h5')
trained_model
```

In Cmd (pip install opencv-python)

```
image = 'MNIST_JPG_testing/1/1038.jpg'

import cv2
from matplotlib import pyplot as plt
img = cv2.imread(image, cv2.COLOR_BGR2GRA
IMAGE_SIZE = 256
img = cv2.resize(img, (IMAGE_SIZE, IMAGE_SIZE))
plt.figure()
plt.axis('off')
plt.imshow(img)

import numpy as np
img = np.array(img)/255.
img = img.reshape((1, IMAGE_SIZE, IMAGE_SIZE, 1))
```

```
[ from Keras.preprocessing.image import
                                ImageDataGenerator

data_generator = ImageDataGenerator(
                    samplewise_center = True,
                    samplewise_std_normalization = True,
                    brightness_range = [0.8, 1.0],
                    zoom_range = [1.0, 1.2],
                    validation_split = 0.1
                 )

[ Img = data_generator.standardize(img)

[ result = trained_model.predict(img)

[ result

[ result.argmax()
```

https:// we.tl/t-2uJILd2PpE

download this file or from WhatsApp
from jabber

Search :- Bootstrap image upload ~~with preview~~

In index.html

```
<body>
    <center>
    <h1> home page </h1>
    <form>
        <Label> select an image </Label> <b
        <input name=" number-image" type="file" > <b
        <button type
```

e