

Main Study Bud

↑ is
≡)

- ① Python -m venv venv
- ② venv\Scripts\activate
- ③ Pip install django
- ④ django-admin startproject studybud
- ⑤ Python manage.py startapp base
- ⑥ add base app to The installed apps:
 - ① base.apps.BaseConfig
 - ② 'base'
- ⑦ Pip install pillow
- ⑧ models.py

Current directory

Ex 2 :-

1. ~~is~~
2. ~~is~~

from django.db import models

from django.contrib.auth.models import

AbstractUser

class User(AbstractUser)

↑
15

User over

① name → CharField → max_length = 200

② bio → TextField → null = True

③ avatar → ImageField → default = avatar.svg, null = True

④ email → EmailField → unique = True, null = True

USERNAME_FIELD = 'email'

REQUIRED_FIELDS = ['username']

⑤ settings.py

AUTH_USER_MODEL = 'base.User'

to app

⑥ makemigration, migrate

- 1) Python manage.py createSuperUser
- 2) class Topic(models.Model)
- ① name → CharField, max_length=200
- Topic Class
- 3) Room
- ① host → ForeignKey (User, on_delete=models.SET_NULL, null=True)
جذر room میباشد topic باید idUser باید باشد
نحوه اینکه null و topic میباشد اما اگر null باشد
آنچه پنهان نماید
- ② topic → n (Topic, n)
- ③ name → CharField, max_length=200
- ④ description → TextField, null=True, blank=True
- ⑤ Participants → ManyToManyField (User, related_name='Participants', blank=True)
وایکی از این دو کاری میباشد
- ⑥ created → DateTimeField, auto_now_add=True
- ⑦ updated → n, auto_now=True
وایکی از این دو کاری میباشد

~~1st~~
9th

class Meta:

ordering = ['-created', '-updated']

(13) Message Model

- ① user → foreignkey (User, on_delete=Cascade)
 - ② room → ↗ (Room, ↘ ↗)
 - ③ body → Text Field
 - ④ created
 - ⑤ updated

def = str = (self);

return self.body[0:50]

۷۸) User Message یا cascade
User ہو تو User گرفتار کر دے message

۱۹:۱
۵ صفحه

۱۴ Make migration, migrate

خطی ۱۰۰

۱۵ base → templates در حافظه
templates → base در داخل فولدر

۱۶ انتقال فایل index.html از فولدر
templates/base در داخل فولدر requirements

۱۷ navbar.html

داخل فولدر templates کے میں navbar.html پر ایک فایل نہ رہے
و سعی دوسرے index.html پر = <header> و جنہیں قرار دیں
<head> { % load static % } </head> و قرار دیں

۱۸ main.html پر ایک فایل templates
, empty body, <head> و ساچل ایک
. CwI Script

وَسَابِقَ

اُورز ۱
صُرُف

{% load static %}

{% include 'navbar.html' %}

بِسْكَرِپْتِ خَالِيَّاتِيِّ بِالْبَدْنِ وَجَرِيدَةِ حَادِثَاتِ

</body> بِلَامِنَتِ مُحَمَّدِيَّةِ وَعَدَازِيَّةِ حَادِثَاتِ

</body>

{% block Content %}

{% endblock Content %}

⑯

index.html

. جَرِيدَةِ الْمَيْدَانِ

{% extends 'main.html' %}

{% block Content %},

<main class="layout layout--3" ...

</main>

{% endblock Content %}

~~لیست~~ ۲۰ setting.py

TEMPLATES = [

'DIRS': [

BASE_DIR / 'templates'

]

۲۱ ایجاد فولدر static در درون دیرکتوری مجموعه و داخل آن فولدرها styles, images و jss را در میان قرار دهید

۲۲ ایجاد فولدر Requirements از فولدر Requirements style.css در میان قرار دهید و داخل فولدر script.js, styles, images در فولدر Requirements/assets از فولدر Requirements/assets icon, images در میان قرار دهید

~~1 je
8)~~

(23) settings.py

STATIC_URL = '/static/'

STATICFILES_DIRS = [

 BASE_DIR / 'static'
]

MEDIA_URL = '/images/'

MEDIA_ROOT = BASE_DIR / 'static/images'

(24) urls.py ←(Project)

from django.conf import settings

from django.urls.static import static

urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

urlpatterns += static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

~~1 jo
9 sp~~ ②5 views.py not home.html -> index.html

②6 views.py

```
def home(request):  
    return render(request, 'base/home.html')
```

②7 create urls.py inside base app
from django.urls import path
from . import views

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

②8 urls.py

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

②9 navbar.html

```

```

~~1, 0
1, 0~~

③ main.html

```
<link rel="shortcut icon" href="{% static  
'images/favicon.ico' %} />  
<link rel="stylesheet" href="{% static 'styles/  
style.css' %} />  
<script src="{% static 'js/script.js' %}">
```

Result

الصورة
ـ إدخالها في المتصفح

④ admin.py

```
from .models import Topic, Room, Message, User  
admin.site.register(User)  
* * {Topic}  
* * {Room}  
* * {Message}
```

~~1 jo
1 jo~~

32 Create another super user called temy
tom@gmail.com

33 Create some Topics

- Topic 1 - Python
- ≈ 2 - JavaScript
- ≈ 3 - Web Development
- ≈ 4 - Django

34 Create some Rooms

RoomName .. Professional coding

- ≈ - Django Lovers
- ≈ - Full Stack Development
- ≈ - Let's try javascript
- ≈ - 300 Days Python

7
Go Liebly

Main Study Bud

2/1
1/1

35 Create Some Messages

36 views.py

حیثیت و وب سایت معرفت

یعنی اسے یہ topic ہے

نامہ ایک میں یافت نہیں کیا جائے اسے ازیز room ہے اسے ایک

خاص ایک room والہ Topic . اسے اسے ایک Room والہ Topic .

topic . room _ set . all . count

~~~  
~~~

درستہ for student اسے Topic ; child کے room

isTopic of home گھر کی سمت میں اور یہ درستہ خاص topic

(new) template ہے topics = Topic . objects . all اور اسے یہ

خاص درستہ topic پر for web (home.htm)

27 براي بحث در فرزنده سواله \rightarrow درستها و آنها
topic خاص و فرزندات آن room ها ساخته شده از آن
topic ها باشند او را حفظ و در رابطه باشند با توجه که مادر اینجا هاست
 $\text{topic.room_set.all().count}$ = عدد فرزند سواله
(topic)

37 برای دیالوگ topic \rightarrow صفحه home.html در

38 topic Filtering

صفحه home میباشد که باید بتواند در آن رسید
که در آن صفحه میباشد و کاربر دارای استر ۹
آن اتفاق نمیافتد \rightarrow برای این استر ۹ \rightarrow topic.name \rightarrow topic
در این استر ۹ نباید برای تمام room ها انتساب شود
View در backEnd \rightarrow request room_message و room
درین استر ۹ فیلتر باید

~~2:0
3:0~~

Topic1-Python $\xrightarrow[\text{ایجاد سیلک}]{\text{سیلک}} \text{url} = 127.0.0.1/39 =$

Topic1-Python

$\langle a \ href = \{\% \text{url} \ 'home' \% \} \? 9 = \{\{\text{topic.name}\}\} \rangle$
 $\{\{\text{topic.name}\}\}$

$\langle \text{lay} \rangle$

: حکمه معرفت گاراپر ۹

$q = \text{request.GET.get('q')}$ if $\text{request.GET.get('q')} != \text{None}$
else ''

: نیزه! سیلک ن اول room بخوبی فیلتر شو

$\text{rooms} = \text{Room.objects.filter}(\text{topic_name_contains} = q)$

: یعنی OR اینه کسی کسی نیزه

from django.db.models import Q

~~2:18
x~~

rooms = Room.objects.filter(Q(topic__

__icontains=q) | Q(name__icontains=q)

| Q(host__username__icontains=q))

room مدلی ۲:۱۹ در topic پلینگ اسلیکس room

۲:۱۹ دلیلی room و User مدلی ۲:۱۹ دلیل

این باره ۲:۱۹ و ۲:۲۰ room و message مودودی

room_messages = Message.objects.filter(room_topic

__name__icontains=q)

بUser برای عوامیو. پرب User برای image میباشد و
اینها در avatar.svg صورت پذیرفتن میباشد

لایسنس جا فلم برای User را سلور و SR لایسنس جا فلم برای.

Be with home www.ps1room.htm) ساعی

نام room وار در صفحه کتاب حاصل می شود . حکایت
مخفی این که کتاب room کتاب حاصل گزینش صورت برداشت می آید :

`room = Room.objects.get(id=pk)`

مکانیزم کے لئے پیکیوں کا درجہ یعنی PK میں room / <str, PK> / کا معنی اسی لئے ہے۔

سر پر request ارسال ہو تو وسیع view دیا جاتے ہیں مگر وہ استمکت room (نیکا بے) کا متعلق ہے۔

2/6

لكل room ينتمي room خاص به يست اورز

جتنى room = مابين room = room
برای این room است اینکه room را در جا بست
room message = room خاص این room ایم لوت کردن
خوبی room message = room.all()

فرزند دار room messages = room.message_set.all()

هر کسی المعرفه نایر کو دید بتواند سود و مادا در آینه هفظ قبل از از دید
که این room خاص باشد

Participants = room.Participants.all() :

③ Copy room.html from Requirements folder to
templates/base folder. We keep just Main Tag
and remove other things

~~مهم~~

Create Room

نحوه ایجاد اتاق جدید و درخواست ایجاد
topic بعنوان سوریه ملکه فحص (ز) Topic
in Topic

Forms.py →

class RoomForm (ModelForm):

model = Room

fields = 'all'

exclude = ['host', 'participants']

لینک label for; create_room ب template.html
و ساخت اتاق ب modelForm ب form ب form
(form) ب برای لینک کار ب {{form.as_p}}
label ب از می شود که برای لینک

3/2

to view \rightarrow مقدمة لـ input

? Context \rightarrow Form = RoomForm()

new label \rightarrow CreateRoom.html \rightarrow Template

\rightarrow name \rightarrow input \rightarrow عنوان المكان

$\{\{form.description\}\}$ $\{\{form.name\}\}$ احرف علامة وبيان description

استطلاع وامانة مور topic اسئلة اسئلة

اسئلة ترى <option>, <select>, <datalist>

احرف عنوان topic \rightarrow input \rightarrow

Value = {{roomTopic.name}} Value

(UpdateRoom() ما يجده كل

{
for topic in Topics
}.

<option> \rightarrow option

<option value = "{{topic.name}}" $\{\{topic.name\}\}$

</option>

fix end for ?

~~3 جزء~~
3 جزء
لـ topic.name \rightarrow option \rightarrow value
لـ topic \rightarrow dropdown list \rightarrow محفوظة ازمه
لـ topic \rightarrow محفوظة ازمه \rightarrow انتخاب

<option value> </option>

= لـ topic \rightarrow قرارداد \rightarrow topic.name \rightarrow محفوظة ازمه

Select \rightarrow لـ topic \rightarrow مارچان \rightarrow Select your topic
and topic \rightarrow انتخاب your topic

def CreateRoom(request):

form = RoomForm()

topics = Topic.objects.all()

if request.method == 'POST':

form = RoomForm(request.POST)

لئے خیلی ممکن:

~~3rd
Q~~

STATIC FILES

۱) settings.py میں

(پہلے اسے کہا جائے) ETL میں (جسے اسے اور دردہ بھرم) -DIRS

load میں static میں style.css میں اوس فارم میں

۲) load میں CSS پر clone کرنا دیکھو

کیا زیرمیں Ctrl+Shift+R کا لامپ کیا؟

کام کیا؟ load کے style.css پر کیا کرنا دیکھو

کیا اسے اس کے ساتھ دیکھا جائے؟

topic_name = request.POST.get('topic')

topic, created = Topic.objects.get_or_create(
name=topic_name)

Room.objects.create(

host=request.user,

topic = topic,
description = request.POST.get('descri-
name = request.POST.get('name')
)
return redirect('home')

Context = { 'form': form, 'topics': topics }

return render (request, 'base/create-room.html'
, context)

Update_room

→ Create-room.html → Upd ①

Topic input to list

Update plus go to value = {{ room.topic.name }}
to Create-room.html message room is edit

3/6)

مودر/ر خا room \rightarrow = subject ~~مودر~~
سوچ \rightarrow update

instance = room

تقریباً topics \rightarrow مودر ارجع topics \rightarrow View معنی
مودر ارجع dropdown list \rightarrow topics

def updateRoom (request, pk):

topics = Topic.objects.all()

room = Room.objects.get(id=pk)

form = RoomForm(instance=room)

if request.method == 'POST':

topic_name = request.POST.get('topic')

topic, created = Topic.objects.get_or_create

form = RoomForm(request.POST, instance=room)

room.name = request.POST.get('name')

room.topic = topic

room.description = \Rightarrow ('description')

~~3:18
7)~~

room. Save()

return redirect('home')

Context = {'room': room, 'topics': topics, 'room': room}

return render(request, 'base/create-room.html', context)

Profile دیگر چیزی نہیں کہ view کرنے والے افراد کو home page پر اپنے topics کا list دیتے ہیں۔

نحو room \rightarrow وزیر \rightarrow User = label UV \rightarrow
User \in deep of room \rightarrow لطف \rightarrow Feed component \rightarrow کوکا

User activity \rightarrow User activity Component 2
new user's user is in room

~~α is
2)~~

rooms = User.^{Parent} room_set.all()

User .^{child} Room \rightarrow User

User to User, User \rightarrow child

room_messages = User.message_set.all()

Message \rightarrow User

Context = { 'topics': topics, 'rooms': rooms, 'room_messages': room_messages }

return render (request, 'base/profile.html', context)

logout, login

from django.contrib import messages

from django.contrib.auth import login, logout

authenticate

~~(x) 3)~~ def LoginPage (request):
 user = request.user
 if request.user.is_authenticated:
 return redirect ('home')
 if request.method = 'POST':
 email = request.POST.get ('username')
 password = request.POST.get ('password')
 try:
 user = User.objects.get (email = email)
 user = authenticate (request, email = email,
 password = password)
 if user is not None:
 login (request, user)
 return redirect ('home')
 else:
 messages.error (request, 'password is not
 correct!')

~~def login(request):~~
~~if request.method == 'POST':~~
 except:

 messages.error(request, 'User does not exist')
 return render(request, 'base/login.html')

def logout(request):
 logout(request)
 return redirect('home')

main.html

</body>

{% if messages %}

<ul class="messages"

{% for message in messages %}

{{ message }}

{% endfor %}

{% endif %} ل. عیار

{% block content %} جزء پایه

{% endblock content %}

~~آیا
جی~~

update user

: یعنی url, برای, b' id و عنوان کی 1 کی
Path ('update-user/'), views.updateUser, name='
update-user'

urls.py → این url میتوان

forms.py

class UserForm(ModelForm):

class Meta:

model = User

fields = ['avatar', 'name', 'username',
'email', 'bio']

edit-user.html

پس از این modelForm بگوییم و ویرایش کردن کنیم
خوب فایل ایجاد کردیم

~~α 18~~

وَارِدَارِ رَاحِدَفْ بَلِيمْ وَمَقْدِيَنْ دَانِخَدَارِ كَبِيلِيمْ رِبَا عَلَفَهُ

نیزهاتاگیع از زیر نمایشگر UserForm و Model Form یا template را در

P. 13.

در حال حاضر edit-user.html نیز را امتحان کنید.

Avatar/Username/email/bio

در حالیکه UserForm ایک سر دہاں میلے گا نہ کوئی دار

Avatar/Name/Username/Email/Bio

↳ \sim to template, also

۲۰۱۷ مارچ edit-user.html

و در نظر داشتم edit-user.html را با Form بسازم

<div class="Form-group" style="margin-bottom: 10px;">Name
Email

<label for="profile-pic"> Avatar </label>

```
<input id="profile_pic" name="profile_pic"  
      type="file" />
```

</div>

۱۰: User میں کوئی صورت میں نہیں۔

```
<Form class="form" action="" method="POST"
```

enctype="multipart/form-data">

{% csrf_token %}

{γ. For field in form γ.}

<div class="form-group">

```
<label for="profile-pic">{{Field-label}}</label>
```

</label>

input voter $\rightarrow \{\{ \text{field} \}\}$

</div>

{% endfor %}

~~(x) 10/10~~

```
def updateUser(request):
    user = request.user
    form = UserForm(instance=user)
    if request.method == 'POST':
        form = UserForm(request.POST,
                        request.FILES, instance=user)
        if form.is_valid():
            form.save()
            return redirect('profile', pk=user.id)
    context = {'form': form}
    return render(request, 'base/edit-user.html',
                  context)
```

url > PK یعنی پیکر کو اینجا در id میگذاریم و اینجا باید پسوند پست نه پست

~~خواهش~~ request.user کاربر کوین کیفیت کیفیت کیفیت کیفیت کیفیت کیفیت کیفیت

{ { room.host.avatar.url } } , live img یو پریس /
- ۱۸۶

~~5~~; ~~ا~~ register user

stable, register <Form> \rightarrow template sign up \rightarrow
div class="form-group" \rightarrow ج \rightarrow ج \rightarrow
{% for field in form %}
<div class="form-group">
<label for=""> {{field.label}}
{{field}} \leftarrow input ج
</div>
{% endfor %}

email, username, name \rightarrow register view: ن
از کاربر این داده را زیرا ذخیره کردن برای پسورد و
عکس و تغییر کاربر کرد که کاربر و کاربر کاربر
و کاربر کاربر کاربر کاربر کاربر کاربر کاربر کاربر

~~in N
2/0~~

urls.py

```
path ('signup/', views.registerUser, name='Signup')
```

forms.py

```
from django.contrib.auth.forms import UserCreationForm
class myUserCreationForm(UserCreationForm):
    class Meta:
        model = User
        fields = ['Username', 'name', 'email', 'password1',
                  'password2']
```

Views.py

```
def registerUser(request):
    form = myUserCreationForm()
    if request.method == 'POST':
        form = myUserCreationForm(request.POST)
        if form.is_valid():
            form.save()
```

5/10
3/3

```
User = form.save(Commit=False)
User.save()
login(request, User)
return redirect('home')

else:
    message.error(request, 'An error occurred
        during registration')

context = {'form': form}
return render(request, 'base/signup.html', context)
```

topic page

given system will be topic or, in programming language
all the topics which are created, can be seen
in home screen, next to topics section button

~~5' il
x) 10~~

URL = pat ('topics/views.topics', name='topic')

views.py

def topics(request):

q = request.GET.get('q') if request.GET.get('q')
l = None else ''

topics = Topic.objects.filter(name__icontains
= q)

context = {'topics': topics}

return render(request, 'base/topics.html', context)

Path ('activity1', views.activitypage,
name='activity')

Recent Activity

3rd
5/10

```
def activityPage(request):
    room_messages = Message.objects.all()
    context = {'room_messages': room_messages}
    return render(request, 'base/activity.html',
                  context)
```

Delete Room

```
Path('delete-room/<str:PK>', views.deleteRoom
      name='delete-room')
```

views.py

```
def deleteRoom(request, PK):
    room = Room.objects.get(id=PK)
    if request.method == 'POST':
        room.delete()
        return redirect('home')
    context = {'obj': room}
    return render(request, 'base/delete.html', context)
```

~~5:10
6:00~~

Back

[write message in specific room]({{request.META.HTTP_REFERER}}></p></div><div data-bbox=)

views.py → room(): ~~return render(request, 'room.html')~~ ^{room()}

if request.method == 'POST':

message.objects.create(

user=request.user,

room=room,

body=request.POST.get('body')

)

room.participants.add(request.user)

→ input name="body" to room.html

~~5.10
+/-~~

delete room messages

urls.py → Path ('delete-message//', views.
deleteMessage, name='delete-message')

views.py

```
def deleteMessage(request, pk):
    message = Message.objects.get(id=pk)
    if request.method == 'POST':
        message.delete()
        return redirect('room', message.room.id)
    context = {'obj': message}
    return render(request, 'base/delete.html', context)
```

(²⁰²⁵ 8/2)

if request.user == message.user: room.html

<svg>... </svg> → <i>{{y.endif}}</i>