

# CS50x Iran

راهنمای پروژه Commerce

HARVARD  
UNIVERSITY



شروع کار

کد توزیع را از

<https://cdn.cs50.net/web/2020/spring/projects/2/commerce.zip>

دانلود کرده و از حالت فشرده خارج کنید

`cd commerce`

با این دستور به دایرکتوری commerce بروید.

با دستور `python manage.py makemigrations auctions`  
تا مهاجرت داده اپ auctions انجام شود.

با دستور `python manage.py migrate` تغییرات رو در دیتابیس اعمال کنید.

#### PROJECT4

##### network

> \_\_pycache\_\_

> migrations

##### static\network

# styles.css

##### templates\network

<> index.html

<> layout.html

<> login.html

<> register.html

\_\_init\_\_.py

admin.py

apps.py

models.py

tests.py

urls.py

views.py

> project4

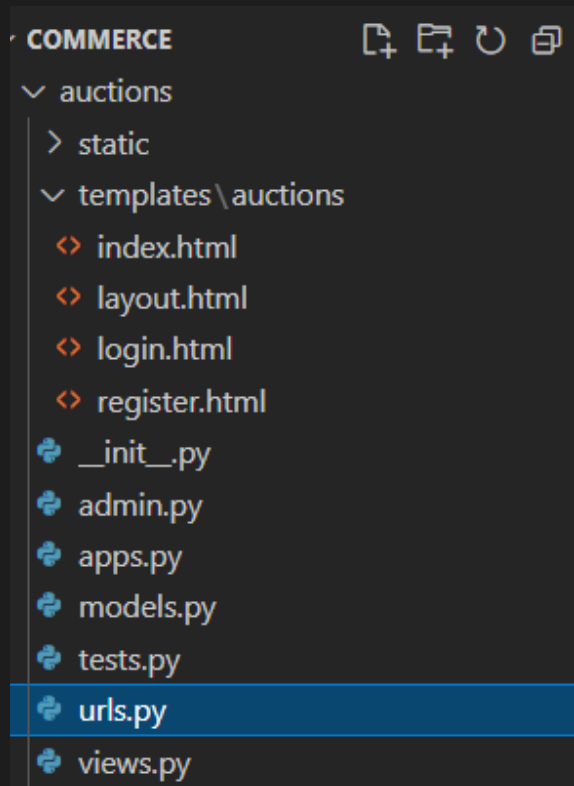
db.sqlite3

manage.py

درک کردن



در کد توزیع، پروژه جنگو به نام commerce داریم که حاوی یک برنامه به نام auctions است.

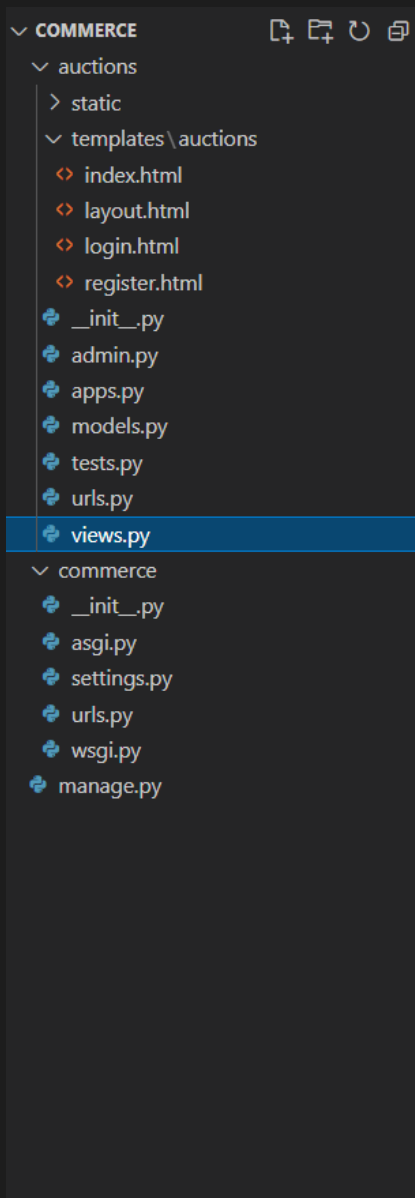


```

auctions > urls.py > ...
1  from django.urls import path
2
3  from . import views
4
5  urlpatterns = [
6      path("", views.index, name="index"),
7      path("login", views.login_view, name="login"),
8      path("logout", views.logout_view, name="logout"),
9      path("register", views.register, name="register")
10 ]
11

```

پیکربندی URL در این فایل  
تعریف شده است .  
همانطور که میبینید چند مسیر  
پیش برایتان تعریف شده است  
و شما باید بسته به نیازتان به  
آنها اضافه کنید.



```
views.py > ...
1  from django.contrib.auth import authenticate, login, logout
2  from django.db import IntegrityError
3  from django.http import HttpResponseRedirect, HttpResponseRedirect
4  from django.shortcuts import render
5  from django.urls import reverse
6
7  from .models import User
8
9
10 def index(request):
11     return render(request, "auctions/index.html")
12
13
14 def login_view(request):
15     if request.method == "POST":
16
17         # Attempt to sign user in
18         username = request.POST["username"]
19         password = request.POST["password"]
20         user = authenticate(request, username=username, password=password)
21
22         # Check if authentication successful
23         if user is not None:
24             login(request, user)
25             return HttpResponseRedirect(reverse("index"))
26         else:
27             return render(request, "auctions/login.html", {
28                 "message": "Invalid username and/or password."
29             })
30     else:
31         return render(request, "auctions/login.html")
32
33
34 def logout_view(request):
35     logout(request)
36     return HttpResponseRedirect(reverse("index"))
37
```

Network/views.py



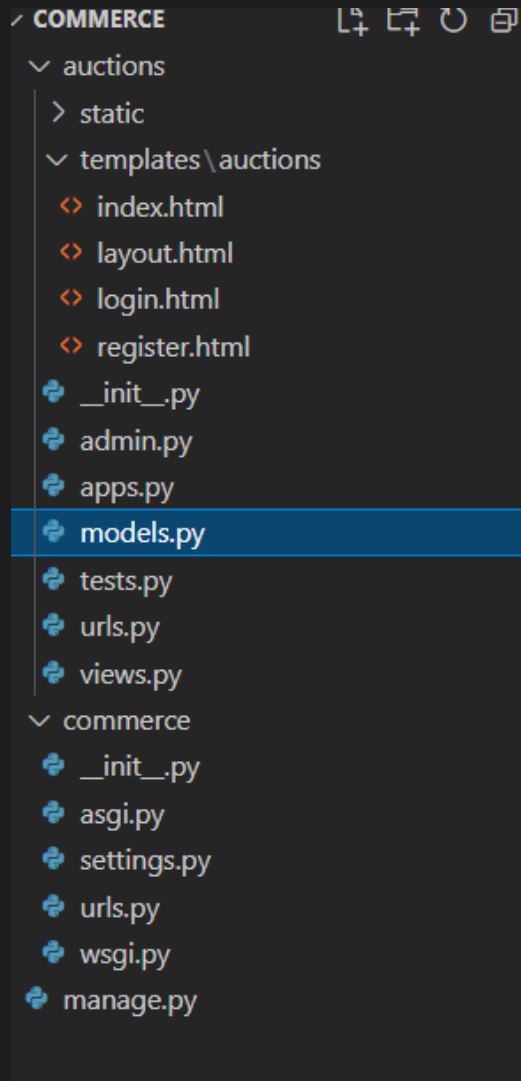
ویو های مرتبط با هرکدام از مسیر هارو  
میتوانید در این فایل مشاهده کنید .  
ویو index فعلا قالب تقریبا خالی  
index.html را برمیگرداند  
ویو login\_view وقتی کاربر دستور GET  
میفرستد ، فرم لاگین رو برای کاربر  
برمیگرداند. وقتی فرم رو با متد POST  
میفرستد، کاربر هویت سنجی میشود، لاگین  
میکند و به صفحه index هدایت میشود.  
ویو logout\_view کاربر را خارج میکند.



- ▼ auctions
  - > static
  - ▼ templates\auctions
    - index.html
    - layout.html
    - login.html
    - register.html
  - \_\_init\_\_.py
  - admin.py
  - apps.py
  - models.py
  - tests.py
  - urls.py
  - views.py
- ▼ commerce
  - \_\_init\_\_.py
  - asgi.py
  - settings.py
  - urls.py
  - wsgi.py
  - manage.py

```
38
39 def register(request):
40     if request.method == "POST":
41         username = request.POST["username"]
42         email = request.POST["email"]
43
44         # Ensure password matches confirmation
45         password = request.POST["password"]
46         confirmation = request.POST["confirmation"]
47         if password != confirmation:
48             return render(request, "auctions/register.html", {
49                 "message": "Passwords must match."
50             })
51
52         # Attempt to create new user
53         try:
54             user = User.objects.create_user(username, email, password)
55             user.save()
56         except IntegrityError:
57             return render(request, "auctions/register.html", {
58                 "message": "Username already taken."
59             })
60         login(request, user)
61         return HttpResponseRedirect(reverse("index"))
62     else:
63         return render(request, "auctions/register.html")
64
```

مسیر register فرم ثبت نام را به کاربر نمایش میدهد ، وقتی فرم ارسال شد کاربر جدیدی ایجاد میکند. تمام اینها در کد توزیع برایتان انجام میشود، بنابراین الان باید بتوانید برنامه را برای ایجاد و لاگین کاربر اجرا کنید .



auctions > models.py > ...

```
1 from django.contrib.auth.models import AbstractUser
2 from django.db import models
3
4
5 class User(AbstractUser):
6     pass
7
```

Network/models.py



هر مدل نشان دهنده نوع داده هایی است که میخواهید در پایگاه داده تان ذخیره کنید. مدل User تک تک کاربران برنامه را نشان میدهد. این مدل از AbstractUser ارث میبرد، که شامل فیلدهایی برای نام کاربری، ایمیل، رمز عبور، و غیره دارد. در صورت نیاز به فیلدهای بیشتر میتونید فیلدهای موردنظرتون رو به مدل اضافه کنید.

مدل های دیگری نیز لازم دارید اضافه کنید که جزئیات فهرست های حراجی، قیمت های پیشنهادی، نظرات، و دسته بندی های حراجی را نمایش دهند.

بعد از اعمال هر تغییری در این فایل باید دستورات  
Python manage.py makemigrations  
Python manage.py migrate  
را اجرا کنید تا این تغییرات به دیتابیس منتقل شود.





مشخصات

برنامه تان باید علاوه بر مدل User حداقل سه مدل داشته باشد: یکی برای فهرست های حراجی، یکی برای قیمت های پیشنهادی، و یکی برای نظرات داده شده درباره فهرست های حراجی. اینکه هر فیلد چه فیلدهایی باید داشته باشد و این فیلدها از چه نوعی باید باشند به شما بستگی دارد. اگر بخواهید، میتوانید مدل های بیشتری داشته باشید.



Title	
Chosse a category:	Jewerly ▼
Description	
\$	Starting Bid
Image URL	
https://example.com/	
<input type="button" value="Submit"/>	

ایجاد لیست

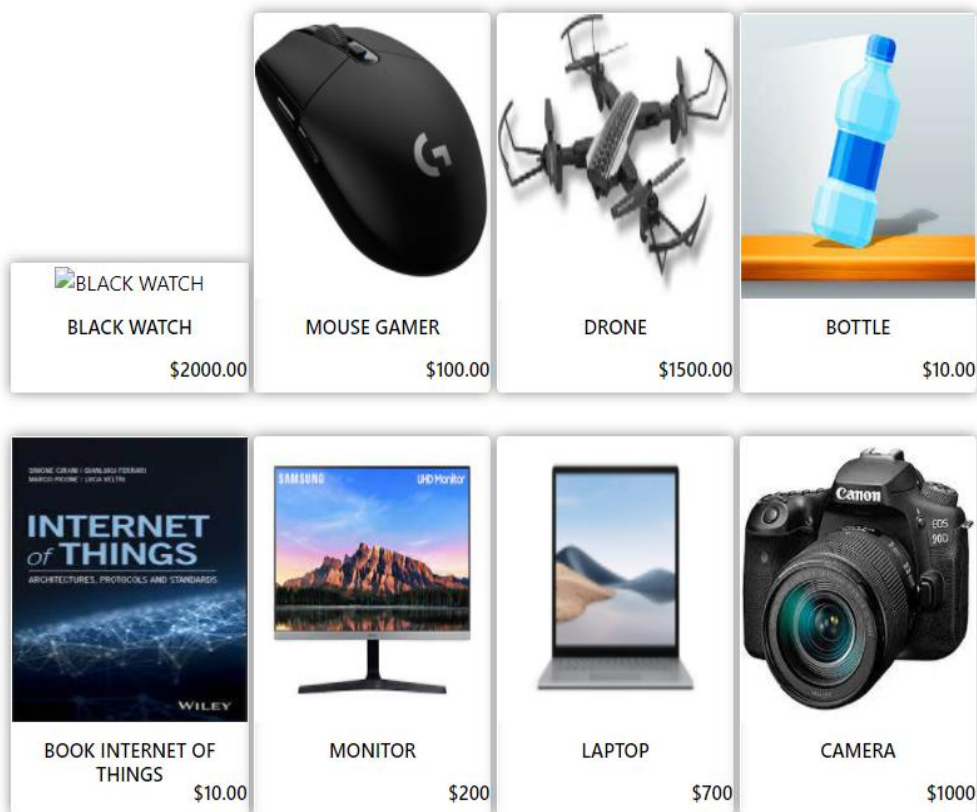


کاربران باید بتوانند به صفحه ای بروند و فهرست جدیدی بسازند. آنها باید بتوانند عنوان آن فهرست، توضیحات متنی، و قیمت پیشنهادی اولیه را مشخص کنند. کاربران همچنین باید بتوانند در صورت دلخواه آدرس URL یک تصویر را برای آن فهرست و / یا دسته بندی(مثل مُد، اسباب بازی، الکترونیک، خانه، و غیره) ارائه کنند .

## صفحه فهرست فعال (Active Listings Page)



مسیر پیشفرض برنامه وب تان باید به کاربران اجازه دهد که تمام فهرست های حراجی فعال فعلی را مشاهده کنند. به ازای هر فهرست فعال، این صفحه باید حداقل عنوان، توضیحات، قیمت جاری، و عکس (اگر عکسی برای آن فهرست وجود داشته باشد) را نمایش دهد.





Listing: Mouse Gamer

Current Bid:

USD 101

Enter USD 101 or more

\$

[Add to Watchlist](#)

### Additional Information

Category	Gamer
Start Price	\$100.00
Description	Pro mouse
Who created	rafael

[Submit](#)

[matheus](#) This mouse is good, you forgot to say that is bluetooth

صفحه فهرست (Listing Page)



کلیک روی فهرست باید کاربران را به صفحه مخصوص آن فهرست ببرد در آن صفحه ، کاربران باید بتوانند تمام جزئیات مربوط به آن فهرست را مشاهده کند، از جمله قیمت فعلی آن فهرست.

اگر کاربر وارد شده باشد، باید بتوانند آیتمی را به "Watchlist" خود اضافه کند. اگر آن آیتم از قبل در "Watchlist" باشد، باید قادر به حذف آن باشد

اگر کاربر وارد شده باشد، باید بتواند برای آن آیتم قیمت بدهد. این قیمت پیشنهادی باید حداقل به اندازه قیمت پیشنهادی اولیه باشد، و باید از سایر قیمت های پیشنهادی که مطرح شده است (در صورت وجود) بزرگتر باشد. اگر قیمت پیشنهادی با این معیارها همخوانی نداشت، باید به کاربر خطا نشان داد



اگر کاربر وارد شده باشد و همان کسی باشد که فهرست را ایجاد کرده است، باید بتواند مزایده مورد نظر را در این صفحه ببندد؛ با این کار، بالاترین پیشنهاد کننده برنده مزایده میشود و این فهرست دیگر فعال نخواهد بود .

اگر کاربر وارد صفحه ای شود که فهرست آن بسته شده است، و اگر برنده آن مزایده باشد، این موضوع باید در آن صفحه عنوان شود.

کاربرانی که وارد میشوند باید قادر باشند در صفحه کالا نظر بگذارند. صفحه کالا باید تمام نظرات ارائه شده در آن فهرست را نمایش دهد



## Watchlist

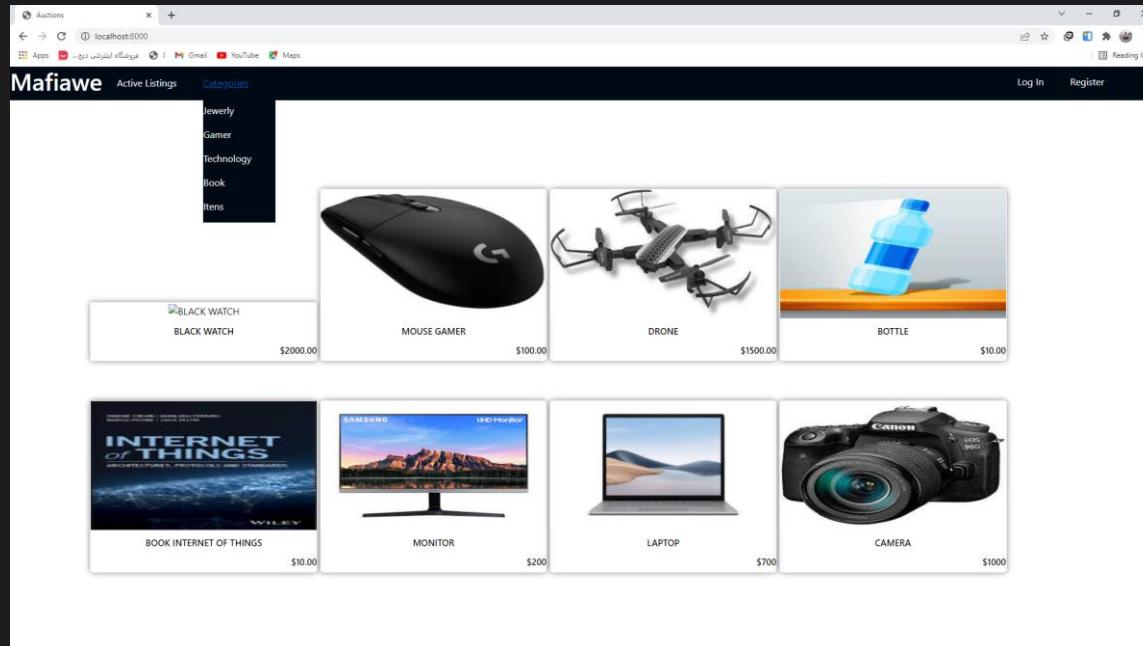


کاربرانی که وارد میشوند باید بتوانند به صفحه Watchlist بروند. این صفحه تمام فهرست هایی را که کاربر به watchlist خود اضافه کرده است نمایش میدهد. کلیک روی هر کدام از این فهرست ها باید کاربر را به صفحه آن فهرست ببرد.

## دسته بندی ها



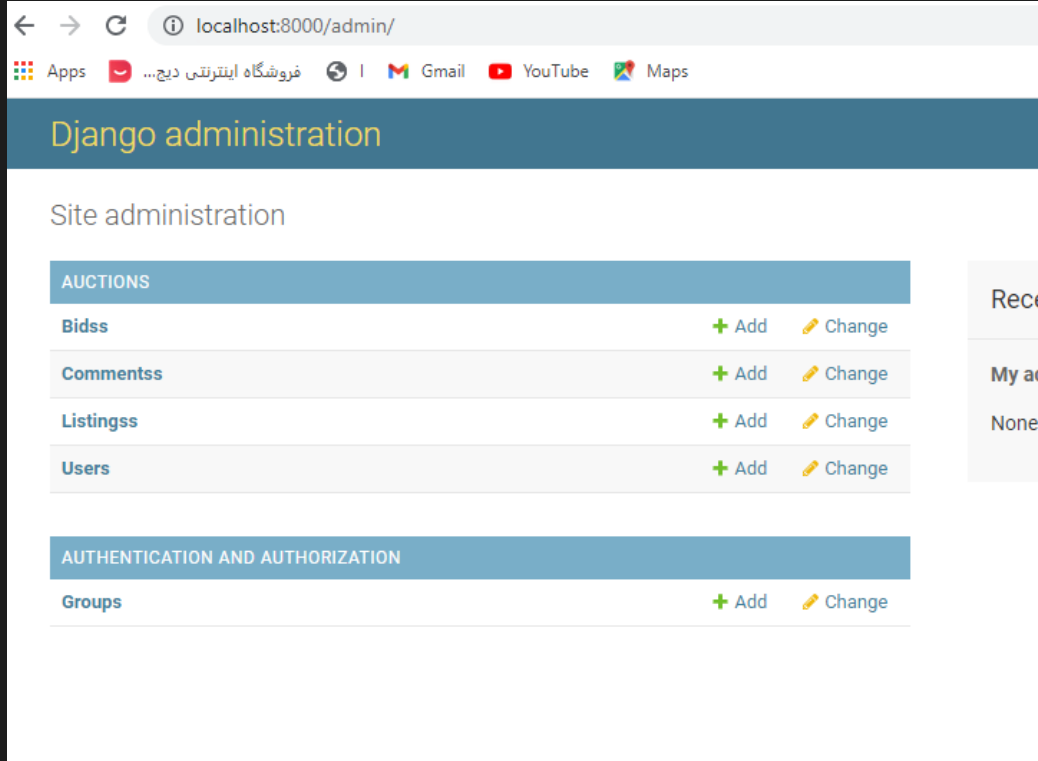
کاربران باید بتوانند به صفحه ای بروند که تمام دسته بندی های فهرست بندی را فهرست وار نمایش میدهد. کلیک روی نام هر کدام از دسته بندی ها باید کاربر را به صفحه ای ببرد که تمام فهرست های فعال در آن دسته بندی را نمایش میدهد.



## رابط مدیر جنگو (Django Admin Interface)



با استفاده از رابط مدیر جنگو، مدیر سایت باید قادر به مشاهده، افزودن، ویرایش، و حذف فهرست ها، نظرات، و قیمت های پیشنهادی ارائه شده در سایت باشد .







برای ایجاد حساب , superuser که به رابط مدیر جنگو دسترسی دارد،  
python manage.py createsuperuser را اجرا کنید.



## نحوه ارسال از طریق CS50 ide

<https://cs50x.ir/winter/static/web/weeks/submit-web-project.pdf>

1- به `ide.cs50.io` بروید و بر روی “Sign in with GitHub” کلیک کنید تا به CS50 IDE خود دسترسی پیدا کنید.

2- فایل اصلی پروژه خود را درون این دایرکتوری آپلود کنید.

3- `cd commerce` را اجرا کنید تا به مسیر پروژه بروید.

4- دستور `submit50 web50/projects/2020/x/commerce` را اجرا کنید تا پروژه هفته هفتم چهارم با موفقیت سابمیت شود.

#submit50 از شما نام کاربری اکانت گیت هابتان و **Personal Access Token** میخواهد که میتوانید طی مراحل ذکر شده نحوه سابمیت کردن (موجود در جلسه صفر) **PersonalAccessToken** خود را دریافت کنید.

در یک ویدیو حداکثر 5 دقیقه ای نحوه کارکرد پروژه خود را نشان دهید و حتما تایم استمپ گذاری کنید.

این فرم را سابمیت کنید



## نکته ارسال

وقتی پروژه‌تان را ارسال می‌کنید، محتوای شاخه `web50/projects/2020/x/commerce` باید با ساختار فایل کد توزیع غیر فشرده، همان‌طور که در ابتدا دریافت شده بود، مطابقت داشته باشد. به عبارتی، فایل‌هایتان نباید داخل هیچ‌کدام از دایرکتوری‌هایی که خودتان ایجاد کرده‌اید بنشینند. شاخه شما همچنین نباید به جز این پروژه، حاوی کد پروژه‌های دیگر باشد. عدم رعایت این ساختار فایل احتمالاً باعث رد شدن کار ارسالی‌تان می‌شود.

برای مثال، در این پروژه، اگر کارمندان نمره‌دهی به آدرس

`https://github.com/me50/USERNAME/tree/web50/projects/2020/x/commerce` بروند

(جایی که USERNAME برابر است با نام کاربری‌تان در گیت‌هاب که در فرم زیر ارائه شده است)، باید دو زیر دایرکتوری (`auctions`، `commerce` و همین‌طور فایل `manage.py` را ببینیم. بعد از بررسی این موضوع، اگر کدتان به این شکل سازماندهی نشده است، مخزن کارهایتان را مطابق با این پارادایم دوباره ساماندهی کنید.



# CS50x Iran

which uses Harvard University's introduction to the  
intellectual enterprises of computer science and the art of programming course



CS50x.ir