



DJANGO TASK 6



Create a new Django project and app?

```
(demoproject) C:\Users\HP\OneDrive\Documents\python demo\Python Projects beinex\demo\demo>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

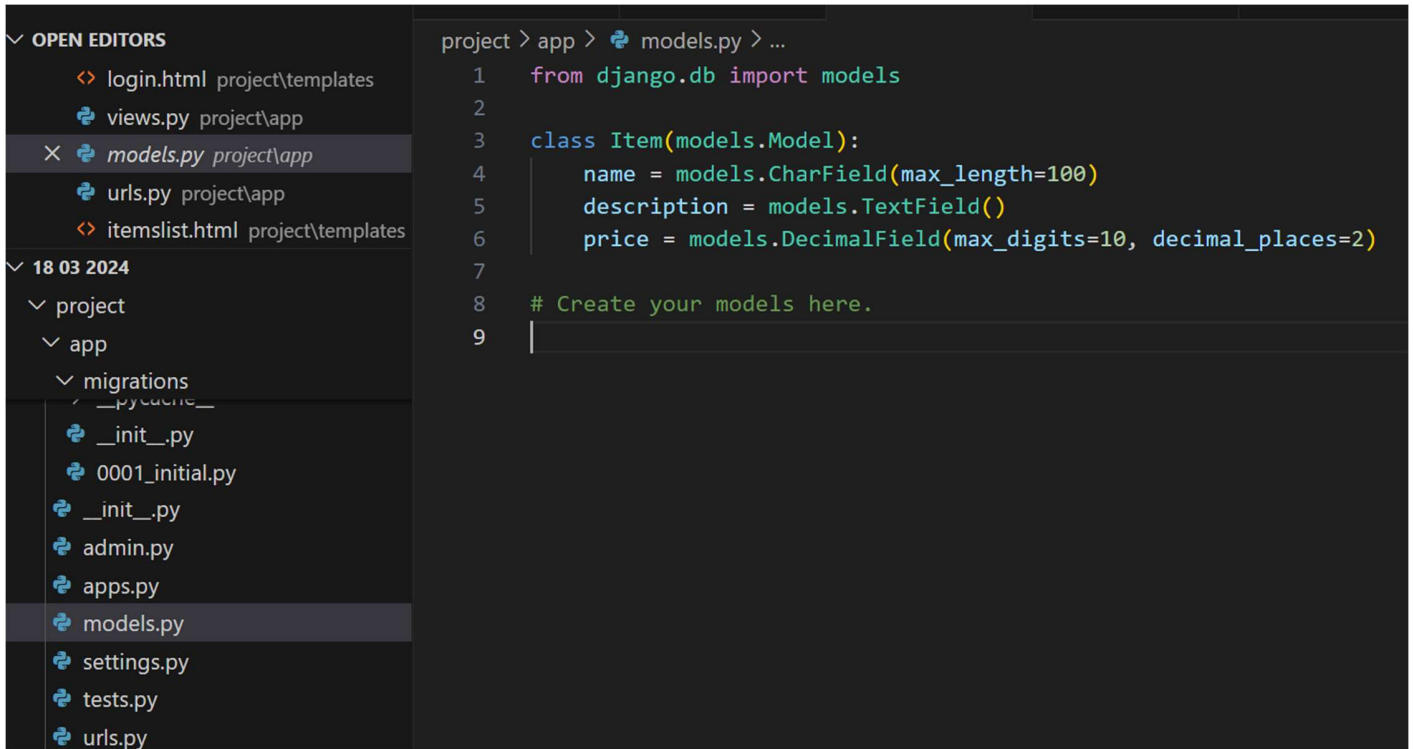
You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 12, 2024 - 11:34:30
Django version 5.0.3, using settings 'demo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

[12/Mar/2024 11:34:33] "GET / HTTP/1.1" 200 10629
Not Found: /favicon.ico
[12/Mar/2024 11:34:34] "GET /favicon.ico HTTP/1.1" 404 2108
C:\Users\HP\OneDrive\Documents\python demo\Python Projects beinex\demo\demo>python manage.py makemigrations
No changes detected

C:\Users\HP\OneDrive\Documents\python demo\Python Projects beinex\demo\demo>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 12, 2024 - 11:50:19
Django version 5.0.3, using settings 'demo.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

Create required apps and models?



The screenshot shows a code editor with a dark theme. On the left is the 'EXPLORER' sidebar showing the project structure. The 'project' folder is expanded, showing subfolders 'migrations' and 'app'. The 'app' folder is expanded, showing files like '__init__.py', '0001_initial.py', 'admin.py', 'apps.py', 'models.py', 'settings.py', 'tests.py', and 'urls.py'. The 'models.py' file is selected. The main editor area shows the code for 'models.py' with the following content:

```
project > app > models.py > ...
1  from django.db import models
2
3  class Item(models.Model):
4      name = models.CharField(max_length=100)
5      description = models.TextField()
6      price = models.DecimalField(max_digits=10, decimal_places=2)
7
8  # Create your models here.
9  |
```

Create views for login, logout and listing items from a model which can be accessed only by logged in users?

```
from django.shortcuts import render, redirect
from django.contrib.auth.decorators import login_required
from django.contrib.auth import authenticate, login, logout
from django.http import HttpResponseRedirect
from .models import Item

def user_login(request):
    if request.method == 'POST':
        user = request.POST.get('username')
        pw = request.POST.get('password')
        user = authenticate(request, username=user, password=pw)
        if user is not None:
            login(request, user)
            success_logins = request.COOKIES.get('success_logins', 0)
            success_logins = int(success_logins) + 1
            response = redirect('itemslist')
            response.set_cookie('success_logins', success_logins)
            return response
        else:
            request.session.setdefault('failed_login_attempts', 0)
            request.session['failed_login_attempts'] += 1
    return render(request, 'login.html', {'failed_login_attempts': request.session.get('failed_login_attempts')})

def user_logout(request):
    logout(request)
    return redirect('login')

@login_required
def item_list(request):
    items = Item.objects.all()
    return render(request, 'itemslist.html', {'items': items})
```

Show the number of failed login attempts in session?



Login

Successful logins: 8

Failed login attempts: 3

```
if user is not None:
    login(request, user)
    success_logins = request.COOKIE.get('success_logins', 0)
    success_logins = int(success_logins) + 1
    response = redirect('itemslist')
    response.set_cookie('success_logins', success_logins)
    return response
else:
    request.session.setdefault('failed_login_attempts', 0)
    request.session['failed_login_attempts'] += 1
return render(request, 'login.html', {'failed_login_attempts': request.session.get('failed_login_attempts')})
```

Show the number of successful login in cookies?

Login

Successful logins: 1

Failed login attempts: 1

```
if user is not None:
    login(request, user)
    success_logins = request.COOKIES.get('success_logins', 0)
    success_logins = int(success_logins) + 1
    response = redirect('itemslist')
    response.set_cookie('success_logins', success_logins)
    return response
else:
    request.session.setdefault('failed_login_attempts', 0)
    request.session['failed_login_attempts'] += 1
return render(request, 'login.html', {'failed_login_attempts': request.session.get('failed_login_attempts')})
```

[YouTube](#) [WhatsApp](#) [Welcome to](#) [python](#) [Introduction to HTML](#) [Create A Respon](#)

Items List

- Apple - apple is sweet - \$100.00

[Logout](#)