Hi, we are building a risk system for a stock portfolio. We have a sample-data.py that gets our sample data and saves it to a csv file and we have a exploratory-analysis.py that does a exploratory analysis on the sample data. We also created a my-functions.py file which will store our functions for the risk system project. Now I want to create a web based application that the user does the login, and if the login is successful he enters on a data input page. I’m doing this in Django and I already have my user model, loginform and want now to add my login view.

To create a login view in Django, you will need to do the following:

1. In your Django project's **urls.py** file, add a new URL pattern for your login view. This will define the URL that users will visit in their web browsers to access the login page. For example:

Copy code

from django.urls import path from . import views urlpatterns = [ path('login/', views.login\_view, name='login'), # other URL patterns... ]

1. In your project's **views.py** file, create a new view function called **login\_view** that will handle the user login process. This function will be called when a user visits the login page.

Copy code

from django.shortcuts import render from django.contrib.auth import authenticate, login from .forms import LoginForm def login\_view(request): if request.method == 'POST': # User has submitted the login form form = LoginForm(request.POST) if form.is\_valid(): # Form data is valid username = form.cleaned\_data['username'] password = form.cleaned\_data['password'] user = authenticate(request, username=username, password=password) if user is not None: # User is a valid Django user login(request, user) # Redirect to a success page else: # User is not a valid Django user # Show an error message else: # User has not submitted the login form form = LoginForm() # Render the login page return render(request, 'login.html', {'form': form})

1. In your project's templates directory, create a new HTML file called **login.html** that will be used to render the login page. This file should contain a form that allows users to enter their username and password, and submit them for authentication.

Copy code

<h1>Login</h1> <form method="post"> {% csrf\_token %} {{ form.as\_p }} <button type="submit">Log in</button> </form>

1. Finally, you will need to make sure that your Django project is configured to use the authentication system. This typically involves adding the **django.contrib.auth** application to your project's **INSTALLED\_APPS** setting, and defining a **LOGIN\_URL** and **LOGIN\_REDIRECT\_URL** in your project's settings.

Copy code

INSTALLED\_APPS = [ # other apps... 'django.contrib.auth', ] LOGIN\_URL = '/login/' LOGIN\_REDIRECT\_URL = '/data-input/'

With these steps, you should have a basic login view that allows users to log in to your Django project using their username and password. Of course, you may need to customize this code to fit your specific needs, but this should give you a good starting point.

I don't have a templates directory, where should I create one?

In a Django project, the templates directory is typically located within your project's main directory, alongside the **manage.py** file and the other top-level directories like **settings.py** and **urls.py**. If you don't have a templates directory in your project, you can create one by creating a new directory called **templates** at the root of your project. Django will automatically look for templates in this directory when rendering pages for your web application.

For example, if your project's directory structure looks like this:

Copy code

my-project/ ├── manage.py ├── my\_project/ │ ├── \_\_init\_\_.py │ ├── settings.py │ ├── urls.py │ └── wsgi.py ├── requirements.txt

You can create a templates directory like this:

Copy code

my-project/ ├── manage.py ├── my\_project/ │ ├── \_\_init\_\_.py │ ├── settings.py │ ├── urls.py │ └── wsgi.py ├── requirements.txt └── templates/

Once you have created the templates directory, you can create your **login.html** file in this directory, and Django will be able to find it when rendering the login page for your web application.

def login\_view(request): if request.method == 'POST': # User has submitted the login form form = LoginForm(request.POST) if form.is\_valid(): # Form data is valid username = form.cleaned\_data['username'] password = form.cleaned\_data['password'] user = authenticate(request, username=username, password=password) if user is not None: # User is a valid Django user login(request, user) # Redirect to a success page else: # User is not a valid Django user # Show an error message else: # User has not submitted the login form form = LoginForm() # Render the login page return render(request, 'login.html', {'form': form}) pylance gives me a identation error for this code, can you find the problem?

!

network error