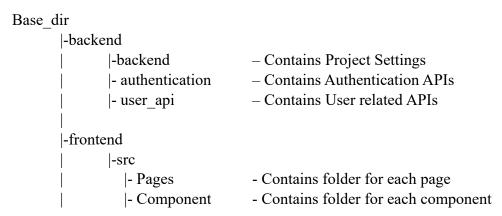
Hiring Task Documentation

Project directory Structure:



Backend:

APIs:

1. Login API

<u> Logiii i i i i i i i i i i i i i i i i i </u>	
<u>URL</u>	http://127.0.0.1:8000/auth/signin/
<u>File</u>	.\backend\authentication\views.py
Request Type	POST
Body	{
	"username": username,
	"Password":password
	}
Input parameters	• username – string
	• password – string
Response	{
	"message": message
	"access": access token
	"refresh": refresh token
	}
Response parameters	message: string -login status
	access: string - JWT access token
	refresh: string - JWT refresh token

2. Sign Up API

<u>URL</u>	http://127.0.0.1:8000/auth/signup/
<u>File</u>	.\backend\authentication\views.py
Request Type	POST
Body	firstName: firstName, lastName: lastName, email: email, username: username, password: password, confirmPassword,

	}
Input parameters	 firstName - string lastName - string email - string - email username - string password - string confirmPassword - string
Response	{ "status": "success" } Or { "error": error message }
Response parameters	status: string error: string

3. Logout API

<u>URL</u>	http://127.0.0.1:8000/auth/logout/
<u>File</u>	.\backend\authentication\views.py
Request Type	GET
Body	-
Input parameters	-
Response	{
	"message": logout status
	}
Response parameters	message: string – success or fail

4. Get CSRF token API

<u>URL</u>	http://127.0.0.1:8000/auth/get csrf/
<u>File</u>	.\backend\authentication\views.py
Request Type	GET
Body	-
Input parameters	-
Response	{'csrfToken': token}
Response parameters	token: string – csrf token

5. Dashboard Data API

<u>URL</u>	http://127.0.0.1:8000/user/dashboard/
<u>File</u>	.\backend\user_api\views.py
Request Type	GET
Body	-
Input parameters	-
Response	{
	'user' : username,
	'watchlists': user watchlists,
	'watchlist companies': companies in watchlist,

```
'allCompanies': companies list,
    'perf_metrics': performance metrics for overall companies

}

Response parameters

User – string
    watchlists – List of Dictionaries of watchlist data
    watchlist_companies – dictionary-
    {
        watchlist name: [companies1, company2], ....
    }

allCompanies – list of Dictionary
perf_metrics – Dictionary
{
        "Max_perf_today": { },
        "Max_perf_10day": { },
        "Min_perf_10day": { },
        "Max_perf_30day": { },
        "Min_perf_30day": { },
        "Min_perf
```

6. Create New Watchlist

<u>URL</u>	http://127.0.0.1:8000/user/addWatchlist/
<u>File</u>	.\backend\user_api\views.py
Request Type	POST
Body	{ name: new watchlist name}
Input parameters	name - string
Response	{'msg':success or failed}
	Or
	{'error': 'Method not allowed'}
Response parameters	message – string
	error - string

7.