

# Hiring Task Documentation

Project directory Structure:

```
Base_dir
|-backend
|   |-backend          - Contains Project Settings
|   |- authentication  - Contains Authentication APIs
|   |- user_api        - Contains User related APIs
|
|-frontend
|   |-src
|       |- Pages      - Contains folder for each page
|       |- Component  - Contains folder for each component
```

Backend:

## APIs:

### 1. Login API

<u>URL</u>	http://127.0.0.1:8000/auth/signin/
<u>File</u>	.\backend\authentication\views.py
<u>Request Type</u>	POST
<u>Body</u>	{ "username": username, "Password": password }
Input parameters	<ul style="list-style-type: none"><li>• username – string</li><li>• password – string</li></ul>
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
<u>Request Type</u>	POST
<u>Body</u>	{ firstName: firstName, lastName: lastName, email: email, username: username, password: password, confirmPassword: confirmPassword, }

	}
Input parameters	<ul style="list-style-type: none"> <li>• firstName - string</li> <li>• lastName - string</li> <li>• email – string - email</li> <li>• username - string</li> <li>• password - string</li> <li>• confirmPassword - string</li> </ul>
Response	<pre>{     "status": "success" } Or {     "error": error message }</pre>
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
<u>Request Type</u>	GET
<u>Body</u>	-
Input parameters	-
Response	<pre>{     "message": logout status }</pre>
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
<u>Request Type</u>	GET
<u>Body</u>	-
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
<u>Request Type</u>	GET
<u>Body</u>	-
Input parameters	-
Response	<pre>{     'user' : username,     'watchlists' : user watchlists,     'watchlist companies': companies in watchlist,</pre>

	<pre>'allCompanies': companies list, 'perf_metrics': performance metrics for overall companies  }</pre>
Response parameters	<pre>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": { },   "Min_perf_today": { },   "Max_perf_10day": { },   "Min_perf_10day": { },   "Max_perf_30day": { },   "Min_perf_30day": { } }</pre>

#### 6. Create New Watchlist

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

#### 7.