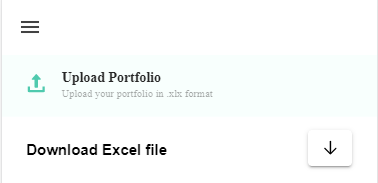
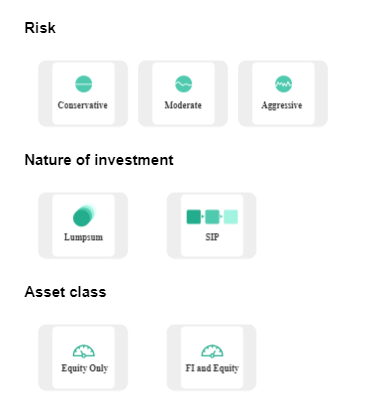
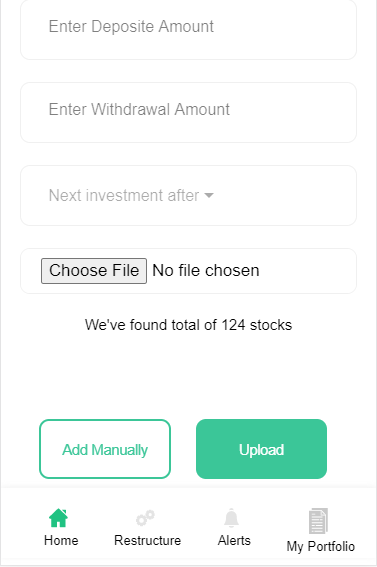
**FUNCTIONAL AND TECHNICAL COMBINED DOCUMENTATION**

**Page (8): Upload Portfolio (Front End Url:** http://localhost:8100/page8/Page8)

**Functional Picture:**







**This Page is for user who is having his existing portfolio.**

**Technical Explanation (Front End and Back End):**

**In Page8.tsx, we are calling following 1 API:**

1. ExistingPortfolioAPI: see backend python file. In this we are doing GET requests.

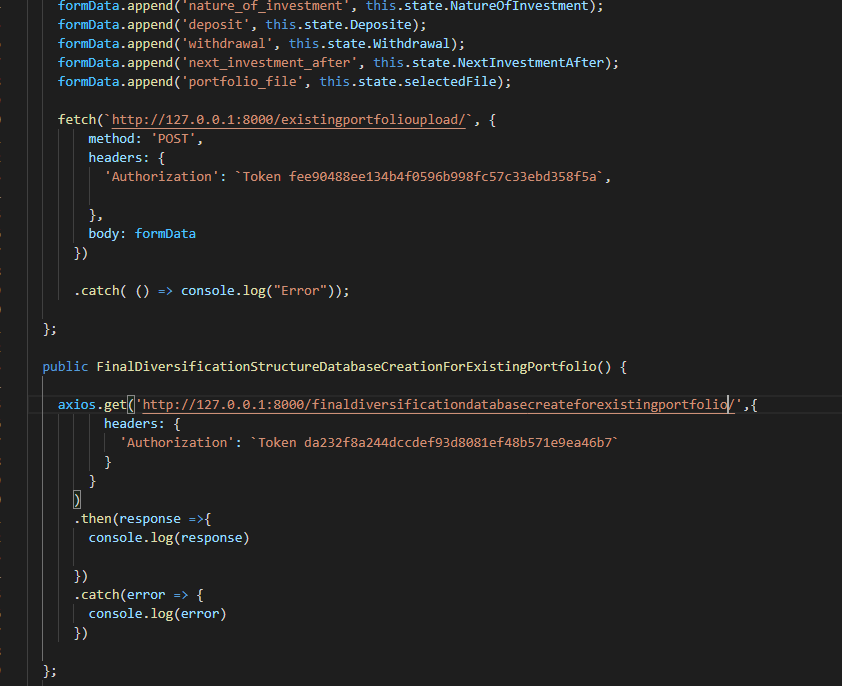


**In the above code snippet, you can see that we have used IonSegment and IonSegmentButton to make the <img> tag clickable and we are saving the click event values in the variables with the help of assignRisk () function. After that we are doing post request with the help of those variables (Conservative, Moderate, Aggressive,) We are doing this for “Risk”, “Nature of Investment” and “Asset class”. Please see the below given code.**



**For “Deposit”,“Withdrawal”, “Next Investment After” and “Upload file” , we are normally putting the value and doing the post request.**

**Now Let’s look at our function which performs the POST request.**



**As you can see above, there is one url:**

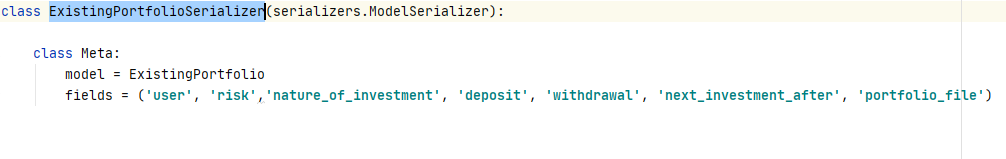
[**http://127.0.0.1:8000/existingportfolioupload/**](http://127.0.0.1:8000/existingportfolioupload/) **--🡪** This url calls ExistingPortfolioAPI which with the help of ExistingPortfolioSerializer

saves the data into the **ClientInformation** model Database in the post method. Please see the below code snippets of **1)** ExistingPortfolioAPI**, 2)** ExistingPortfolioSerializer **and 3)** ExistingPortfolio

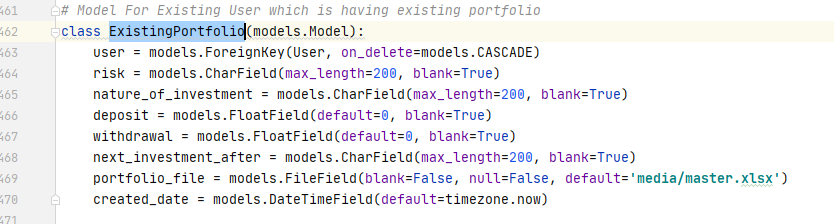
**model. Simultaneously we are saving the portfolio parameters in FinalDiversificationStructure database model.**

Please see the ExistingPortfolioAPI in the backend. In this we are calculating the total investment of the uploaded portfolio by taking the sum of product of stock price and its quantity.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



**DONE!!!**