Project Assessment:

EvaExchange is an arbitrarily trading game developed by a startup in a very short span of time called “Super Traders” . The purpose of the application is to educate users on the terminology used in trading of shares.

Notes:

- The project registers share and and allows users to update the price of the share on an hourly basis; the share registered should have a unique Symbol to identify it and should be all capital letters with 3 characters. The rate of the share should be exactly 2 decimal digits.
- Also, the users should have a portfolio before they can start trading in the shares.
- The frontend application is excluded from the current scope. It is a separate, fully-functional application handled by another team, and we do not want to modify it.

Tasks:

1) For a given portfolio, with all the registered shares you need to do a trade which could be either a BUY or SELL trade. For a particular trade keep following conditions in mind:

BUY:
a) The rate at which the shares will be bought will be the latest price in the database.
b) The share specified should be a registered one otherwise it should be considered a bad request.
c) The Portfolio of the user should also be registered otherwise it should be considered a bad request.

SELL:
a) The share should be there in the portfolio of the customer.
b) The Portfolio of the user should be registered otherwise it should be considered a bad request.
c) The rate at which the shares will be sold will be the latest price in the database.
d) The number of shares should be sufficient so that it can be sold.

Hint: You need to group the total shares bought and sold of a particular share and see the difference to figure out if there are sufficient quantities available for SELL.

2) Please keep in mind following instructions during developing your application.

a) Create a RESTful API for BUY and SELL operations. We have already developed a front-end application. We just need a back-end application (RESTful API).
b) Try to use an ORM tool such as Sequalize ((prefered)), Prisma etc.
c) Try to use a relational database such as PostgreSQL (prefered), MySQL etc.
d) NodeJS is prefered but if you have no experience with NodeJS, you can create RESTful API with Java or .NET Core
e) Initialize your database with BULK INSERT/UPDATE operations by using the selected ORM’s feature. For example, create 5 clients, create a BUY/SELL transaction log for trading operations of each client.
f) Create and share with us a POSTMAN collection to test your API. Create a SELL and BUY end-point calls."

Cheers,