**Completed project is uploaded in GitHub**

https://github.com/suneetha-p/stockpriceDisplay

**Used Skills in this project:**

Java, SpringBoot,Spring Data JPA,H2 database,Maven, Junit,Mockito, GitHub, SpringBoot Embedded Tomcat Server, intelliJ,visual studio, reactJS,PostMan

**What are the task completed in this project:**

• Created stock price API which running on default port 8080 and uses embedded database h2.

• Created front end page using react to display stock pricing table .It is running on default port 3000.

**Brief Explanation about How completed this project:**

• Created SpringBoot API to get data from database and feed it to front end.

• Database table script is in schema.sql file.It will create table with column names id,stock\_symbol,askprice,bidprice,eventime.

• Data will be inserted into table from data.sql file while starting up the application.

• PriceController class has two endpoints

• 1)GetMapping —It will interact with DAO layer and will bring stock\_symbol,avg(bidprice+askprice),trend columns using “WITH CLAUSE SQL” .

• 2)PostMapping—-It will take Json record in body and will save it into database.

• Written junit test cases and results are success.

• Front end pages are developed in react js.

• 1)Table.js file in component folder contain table creation JSX

• 2)StockPrice.js contains a call for RestAPI to get the data for displaying in table.

• 3) call is made every one second to make page reactive.

**How to run projects:**

• Download Stock-price project from GitHub

• Update maven project and resolve usual setup errors if any

• Right click this project and Run As > Spring Boot App

• Once successfully started , then goto browser and hit url http://localhost:8080/h2-console

• Initially it will display login screen and click on connect button.It will show up the Stock\_price table with default data.

• Download stockprice-UI from GitHub.

• run ui project from command line using npm start

• Once successfully started it will open the browser http://localhost:3000/ and will display the table with corresponding column values .

• To test whether page is reactive or not ,using split window feature open postman and browser with url:http://localhost:3000/.

• open the POST Man and make post call with uri: createStockPrice and request body should have {"stockSymbol":"D05:sgx", "bidPrice":"1","askPrice":"111"} and click on send data

• Next it will call postMapping method and will insert data into embedded database and will update UI with latest Trend value.