System Requirements Specification Index

For

e-Stock Market Application

Version 1.0

IIHT Pvt. Ltd.
IIHT Ltd, No: 15, 2nd Floor, Sri Lakshmi Complex, Off MG Road, Near SBI LHO,
Bangalore, Karnataka – 560001, India
interest@iiht.com

STOCKMARKET APPLICATION

System Requirements Specification

1. Project Abstract

StockMarket Application is a simple Spring boot application with MongoDb/MySQL, where it allows any unregistered users to manage the stocks at any stock exchanges like create, view, modify and delete stock price details and company details.

Every **Company Details** should have the properties like company code, company name, company CEO, stock exchange, turnover, board of directors and company profile.

Every **Stock Price Details** should have the properties like company code, post id, current stock price, stock price date, and stock price time.

The relationship between these two entities is that every company would have multiple stock prices and can generate max, min and average stock prices between the stipulated time period.

2. Visitors can perform the follow actions

- Allows to add a new company details and a new stock price detail
- Allows to delete an existing company or existing stock
- Allows to search the company or stock on the basis of company code
- Allows to display all company information or all stock detail
- Allows to display max, min and average stock prices between the stipulated time period.

Toolchain SpringBoot, RESTful Web Services, MySQL.

3. Assumptions, Dependencies, Risks / Constraints

- While adding a company details, if company code is already existing, it should throw a custom exception
- While deleting the company, ensure that company code already exists, if not, the operation should throw a custom exception
- For all rest endpoints receiving @RequestBody, validation check must be done and must throw custom exception if data is invalid
- Must not go and touch the test resources, as they will be used for Auto-Evaluation
- All the business validations must be implemented on DTO object only
- All the database operations must be implemented on model / entity object only
- The conversion from model to dto can be done in the service layer or at separate package
- The methods required for the conversion are provided in the StockMarketUtility class in the 'utils' package in the project.
- The Company Details and Stock Price Details are connected through a field company code – applying integrity constraint

4. URL's Exposed in the controller

CompanyInfoController

URL Exposed	Purpose
/company/addCompany	Add a new Company Details
/company/deleteCompany/{companyCode}	Delete Company with given Company Code
/company/getCompanyInfoById/{companyCode}	Fetches the Company Details with the given Company Code
/company/getAllCompanies	Fetches all the Company Details

StockPriceController

URL Exposed	Purpose
/stock/addStock	Add a new Stock Price Details
/stock/deleteStock/{companyCode}	Delete Stock with given companyCode
/stock/getStockByCompanyCode/{companyCode}	Fetches the Stock with the given companyCode
/stock/getAllStock	Fetches all the Stock Price Details
/stock/getStockPriceIndex/{companyCode}/{startDate}/{endDate}	Fetches Stock Price Index with companyCode and duration

5. Business Validations

- Company Code must be not null and unique
- Company Name is not null, min 3 and max 100 characters.
- Company CEO is not null, min 5 and max 100 characters.
- Company Turnover is not null, precision 10 and scale 2.
- Company Board of Directors is not null, min 5 and max 200 characters.
- Company profile is not null, min 5 and max 255 characters.
- Stock Exchange is not null, min 3 and max 100 characters.
- Current Stock Price is not null, precision 10 and scale 2.
- Stock Price Date is not null and never exceed current date
- Stock Price time is not null and never exceed current time

6. Resources Available in the platform

Spring Boot Rest Package Structure		
NAME	**RESOURCE**	**REMARKS**
controller	CompanyInfoController, StockPriceController	● A ComapnyInfoController class to handle all requests regarding company information ● A StockPriceController class to handle all requests regarding Stock transactions
dto	CompanyDetailsDto, InvalidCompanyExceptionResponse, StockPriceDetailsDto, InvalidStockExceptionResponse	● CompanyDetailsDto: DTO class, InvalidCompanyExceptionResponse: For returning response in case of exception ● StockPriceDetailsDto: DTO class, InvalidStockExceptionResponse: For returning response in case of exception
exception	InvalidCompanyException, CompanyNotFoundException, InvalidStockException, StockNotFoundException	Custom Exception classes
model	CompanyDetails, StockPriceDetails	Entity / Document classes
repository	CompanyInfoRepository, StockPriceRepository	Repository interface declarations
Services	CompanyInfoService, CompanyInfoServiceImpl, StockMarketService, StockMarketServiceImpl	Service Layer Implementations
utils	StockMarketUtility	Class containing resources for inter conversion between model and dto objects

Note

1. **Need to implement ONLY controller and service methods in**
   - **CompanyInfoController.java and StockPriceController.java classes along with**
   - **CompanyInfoServiceImpl.java and StockMarketServiceImpl.java classes**
2. **DO NOT ADD ANY method or CHANGE any Rest End-Point pattern.**

7. Execution Steps to Follow

1. All actions like build, compile, running application, running test cases will be through Command Terminal.
2. To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal.
3. To build your project use command:
   mvn clean package -Dmaven.test.skip
4. To launch your application, move into the target folder (cd target). Run the following command to run the application:
   java -jar StockMarket-0.0.1-SNAPSHOT.jar
5. This editor Auto Saves the code.
6. If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use CTRL+Shift+B-command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
7. These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
8. To test any Restful application, the last option on the left panel of IDE, you can find ThunderClient, which is the lightweight equivalent of POSTMAN.
9. This is a web-based application, to run the application on a browser, use the internal browser in the workspace. Click on the second last option on the left panel of IDE, you can find Browser Preview, where you can launch the application.
   Note: The application will not run in the local browser
10. Default credentials for MySQL:
    a. Username: root
    b. Password: pass@word1
11. To login to mysql instance: Open new terminal and use following command:
    a. sudo systemctl enable mysql
    b. sudo systemctl start mysql
    c. mysql -u root -p
       The last command will ask for password which is 'pass@word1'
12. Mandatory: Before final submission run the following command:
    mvn test

13. **You need to use CTRL+Shift+B - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.**